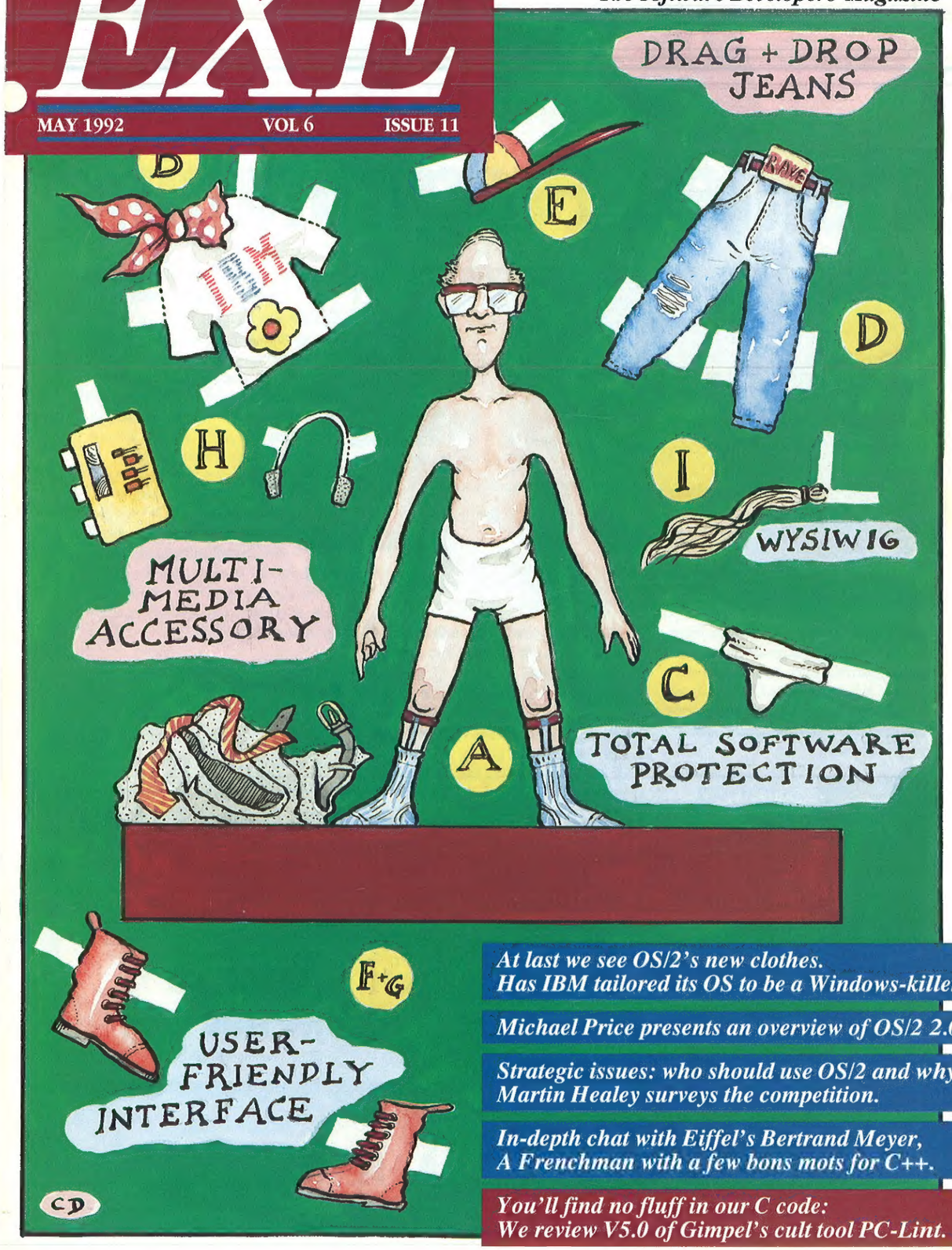
£3.20

EXE

The Software Developers' Magazine

MAY 1992 VOL 6 ISSUE 11



At last we see OS/2's new clothes. Has IBM tailored its OS to be a Windows-killer?

Michael Price presents an overview of OS/2 2.0.

Strategic issues: who should use OS/2 and why? Martin Healey surveys the competition.

In-depth chat with Eiffel's Bertrand Meyer, A Frenchman with a few bons mots for C++.

You'll find no fluff in our C code: We review V5.0 of Gimpel's cult tool PC-Lint.

B O R L A N D

The Leader in Object-Oriented Programming

Professional C programmers are choosing object-oriented C++ because they know it increases productivity. And the C++ they're choosing is from Borland. With more than 500,000 copies of C++ shipped, Borland sets the standard.

Since 1983, Borland has consistently provided professional software development tools to programmers around the world. Today, Borland is the acknowledged leader in Object-Oriented Programming (OOP). OOP gives you amazing code reusability, extensibility and easier maintenance because applications are built on a basis of tested, reliable code.

Borland C++ 3.0. C++ you can rely on!

Borland® C++ is the choice of professional C *and* C++ programmers for Windows *and* DOS application development. The third-generation C++ compiler fully supports the AT&T CFRONT 3.0 standard C++, including templates! And the powerful C compiler fully implements the ANSI C standard.

With both compilers included you can program in C today and move to C++ when you're ready.

A host of powerful features in one package!

Borland C++ comes with an array of powerful yet easy-to-use features demanded by today's professional programmers. The **Windows-hosted IDE** provides a superior software development environment. Now you create, edit, compile and run Windows applications all from within Windows. All resulting in remarkable productivity.

C and C++ for Windows and DOS all in one package!

Now in its third generation! Tested, proven, reliable!

Only Borland C++ gives you the graphical **ObjectBrowser.**™ So you get a visual display of the relationships between objects, and can easily navigate through the classes, functions or variables in your code.

The **EasyWin™ Library** converts standard DOS applications to Windows just by recompiling. The **SpeedBar**™ makes Windows development astonishingly intuitive and fast, employing recognizable icons to represent your frequently used menu items.

Precompiled headers give you quick compilation and Borland C++'s DPMI support lets you compile huge applications fast. **Resource Workshop** allows you to visually create your Windows user interface without programming.

Quality tools that other compilers can't match

Borland C++ also comes complete with all the tools that the serious professional developer needs. Borland's award-winning **Turbo Debugger®** gives you intelligent, interactive debugging on a single monitor. The **Turbo Profiler**™ helps you spot bottlenecks in your code to streamline your application's performance. With the object-oriented **Turbo Assembler®** you can integrate your assembly language code into your applications quickly and easily.

Only Borland's C++ gives you Application Frameworks for DOS and Windows

Borland C++ & Application Frameworks slashes development time by giving you intuitive, ready-made user interfaces for Windows and DOS. Simply plug an application framework into your application or customize it any way you wish. High-level objects are ready to bolt on with just a few lines of code. Add an editor in just one line! Automatically inherit windows, dialogues, menus, mouse support and more.

Borland C++ includes

- C++ with templates, compatible with AT&T CFRONT 3.0
- ANSI C
- Global optimizer, including
  - Global register allocation
  - Local and global common sub-expressions
  - Induction variables
  - Loop and jump optimization
  - Register parameter passing
  - Ten other optimizations
- Fast compile speed
- Windows and DOS integrated development environments
- DPMI support for compiler and environments
- Visual ObjectBrowser that navigates through your code
- EasyWin Library for converting DOS programs to Windows
- Resource Workshop to create a complete range of Windows resources
- WinSight™ message tracking utility
- Turbo Debugger for DOS and Windows
- Turbo Profiler for DOS and Windows
- Object-oriented Turbo Assembler

Borland C++ & Application Frameworks also includes

- ObjectWindows for Windows
- Turbo Vision for DOS
- Complete source code for both Application Frameworks and the runtime library

1991 BYTE AWARD OF DISTINCTION

PC WEEK LABS TOP PRODUCTS OF 1991

#1 Tied in Customer Satisfaction Among Business Users*

J.D. POWER AND ASSOCIATES

1. **Borland**
1. WordPerfect
3. Lotus
4. Microsoft
5. Claris
6. Aldus

The **ObjectWindows**™ application framework streamlines Windows development. It automates initialization and gives your applications more functionality with less code.

ObjectWindows also features Borland Custom Controls that make your applications look great.

Also included is **Turbo Vision**,™ the application framework for DOS that gives you capabilities similar to ObjectWindows for character-based programs. Both come with full source code.

The critics' choice

"Borland C++ looks set to become the *de facto* development environment for Windows programmers."

—*Program Now*, December, 1991

"There are many new features packed into this release . . . Microsoft® has a lot of catching up to do."

—*EXE* magazine, December 1991

Borland makes your transition from C to C++ easy!

To make your transition from C to C++ as smooth as possible, we've created *World of ObjectWindows for C++* training videotape. And now you can get it FREE (retail value £99.95)! This remarkable video training package comes complete with two videos, workbook and sample source code that you can plug right into your program! Taught by internationally recognized C++ experts, *World of ObjectWindows for C++* will give you an in-depth understanding of the ObjectWindows application frame-work for Windows. Combined with Borland C++ & Application Frameworks, there is no faster, more productive way to get up to speed on C++.

If you're serious about moving to C++, go with the leader, Borland.

CIRCLE NO. 902

FREE Video Training Package!

Borland is making your move to C++ as easy as possible with a special, limited-time offer. From now until 1 October 1992, you may redeem this official form for a free *World of ObjectWindows for C++* training package. Just purchase a copy of Borland C++ & Application Frameworks through your local dealer. Then send a copy of your invoice and your registration card, along with this official form, to:

Borland International (U.K.) Ltd.
Freepost RG1571
Twyford, Berkshire RG10 8BR

See your dealer today!

Or call Freefone 0800 212727

*J.D. Power and Associates 1991 Computer End User Satisfaction Study for Application Software Publisher; Phase IV: Business End User Summary. Responses from Business End Users at 4,396 business sites.

B O R L A N D

Vol 6 Issue 11

.EXE

May 1992

General

.EXE Magazine is independent and not affiliated to any vendor of hardware, software or services. It is published by Process Communications Ltd, 10 Barley Mow Passage, Chiswick, London W4 4PH.
Tel: (Advertising/Editorial/Production) 081 994 6477
(Subscriptions) 0442 824501
Facsimile: 081 994 1533 Telex: 8811418 SPACES G
ISSN: 0268-6872

Subscriptions

.EXE Magazine is a monthly journal for software developers. It is available only by subscription, at a cost of £35 per annum (11 issues) in the UK, £50 for 2 years. Overseas subscriptions cost £45 per annum, £70 for 2 years. 'A subscription implies that this journal will be sent to the subscriber until one of the three expires' - AG Macdonell. The magazine is published around the 26th of the month preceding the cover date. There is no January issue. To subscribe or obtain details of overseas/academic rates, please call 044282 4501, or write to The Subscriptions Manager, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We can invoice your company or take ACCESS and VISA cards. In case of a query about your subscription, please call 044282 4501.
Back issues are available at a cost of £3.50 each - call our Chiswick office (081 994 6477) for a list of issues still in print.

Editorial

Editorial enquiries should be addressed to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We welcome letters, opinions, suggestions and articles from our readers. Please write for a copy of our Contributors' Guide.
Information contained in .EXE is believed to be correct. If errors are found, we will endeavour to publish a clarification in the next available issue.
From time to time, we offer to copy the PC program code described in an article onto diskette for our readers. In this case, please send a blank, formatted disk with a self-addressed, prepaid mailer to the editorial address given above. We can copy both 5.25" and 3.5" disks.
The publishers can accept no liability for any consequences of using software distributed in this way.

Advertising

If you are interested in advertising in this magazine, please write to the Display Advertising Manager/Recruitment Sales Manager (as appropriate), .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH, or call 081 994 6477 for details of our advertising rates.

Pronunciation

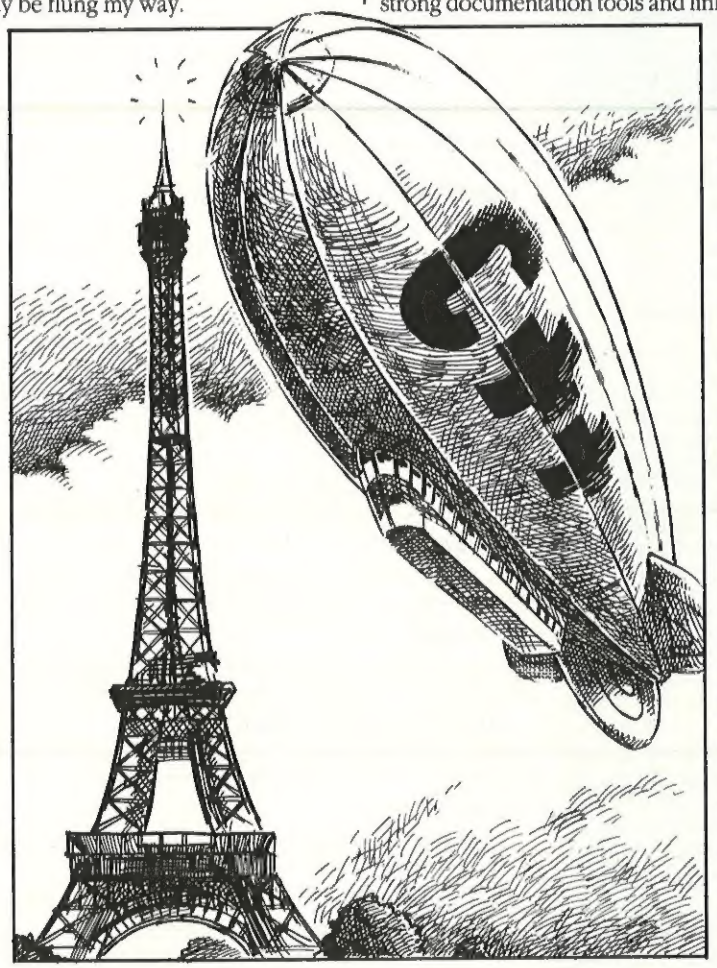
.EXE Magazine rhymes with 'not sexy magazine'.

Issue theme: OS/2

Oh no, not again...

At the beginning of the 1980s, Pascal and C slugged it out for the title Best General Purpose Programming Language. Will Watts believes the battle is about to be re-fought.

Of course, you might view it as a piece of pure cynicism on the part of .EXE Magazine. There can be no doubt that quarrels over methodology, especially language quarrels, are the very meat of readable editorial. (Quiche eater! Assembler programmer!) However, there is a battle hotting up, virtually a rematch, which is now beginning to spill out of the towers of academia into real life, and I believe that its importance to software development is sufficient to quash any accusations of warmongering that may be flung my way.



In the blue corner, then, we have Bjarne Stroustrup's C++. This is the conservative choice, and must be the general current favourite to win. With all the major compiler vendors either offering or promising C++, its impact on the market is assured. There are plenty of C programmers waiting to be post-incremented (I know it's not that simple, but that is the deepest level of understanding you'll get out of the average recruitment consultant, which may be what matters) and the machinery of standardisation has long since been set in motion.

Opposition to C++, it seems to me, comes in the form of two main school of thought (not necessarily mutually exclusive). There is the camp which never cared for C anyway - the 'quiche eaters'. These people never acknowledged that the flexibility of C justified the dangers that its use introduced. To them, C++ is even more of a monster; a few of the more gross type problems cleaned up, perhaps, but the potential to create cataclysmic cock-ups from gargantuan, incomprehensible multiply-inherited hierarchies with a confusion of pointers and references exploding in all directions.

The other class (sorry) of anti-C++ talk comes from the object-oriented purists. Briefly, this camp hates C++ for its mixture of procedural and OOP programming. Their slogan could be: 'Do it properly or let it alone!'

Where should these malcontents hang their standard? OOP pioneers LISP and Smalltalk have established their niches, but the consensus is that they are going to stay in them. They are just too wacky for their own good.

Alternatively: every manufacturer of 3GL or 4GL languages is currently nailing on OOP extensions; we can expect COBOL and Basic any moment now, and I dare say in some distant garret even dear old FORTRAN is getting the treatment. But languages created in this way must suffer from the same lack of 'purity' as C++ itself, and, what's more, are unlikely to have benefited from the same careful engineering as Stroustrup's team has put in. No help from the old Pascal/Modula master Niklaus Wirth; he has unusual views regarding what OOP is about *(see .EXE May 1990)* and is isolated from the mainstream.

So we come to Eiffel. Meyer's language is a 'clean' OOPL, like Smalltalk, and has an associated programming environment. Unlike Smalltalk, it compiles and links like a 'proper' language. On the page, it reads like a Pascal variant - it doesn't require you to abandon all the programming conventions you have ever learned. It has a library, and strong documentation tools and linkage to existing C code. It is strictly typed in the Wirthian manner, but avoids many of those tiresome re-declarations that you find in Ada and Modula-2. It has a fistful of extensions to the basic OOP ideas. In short, it's pretty *neat.*

What is the C++ camp's view of Eiffel? About a year ago, I mentioned to Stroustrup *(Reader's voice: 'Ooh, you disgusting name-dropper!')* that I admired Meyer's first Eiffel book, and wondered if this wasn't the way to go. Stroustrup shook his head and said mildly: 'Go away, try solving a real problem in Eiffel, and see what you think.' Which I interpret to mean that Eiffel contains impracticalities that you only hit when you get into real-life problems.

This, of course, is very close to the old argument offered by C proponents against Pascal. Pascal was great - provided you didn't need to pass variable length strings to a procedure, or use files, or create multi-module programs, or get at low-level machine features. All Pascal's major problems (except not being the basis of UNIX) were eventually solved by proprietary implementations and by the Modulas - but by then it was too late.

The practical problems of C++ are now emerging rapidly from the woodwork under the sheer pressure of the huge numbers using it. They are not blaring faults, as were found in the intended-for-teaching-only language Pascal, but they are there, and they are often solved by the design of Eiffel. But then Eiffel is mostly untested, so...

In the end, the ideal thing to do is to take Stroustrup's advice, and find out for yourself. (I have not, I must admit guiltily, yet been able to do this - the only Eiffel implementation ever seen in this office is currently out for review. But I will.) Most likely that is a tall order for most of our readers, in which case this piece serves simply to alert you to the fact that there is still life outside the *Sturm und Drang* of C++.

EXE

To find out more about current implementations of Eiffel, you should contact Applied Logic Distribution (081 780 2324).

GREY MATTER

Prigg Meadow, Ashburton, Devon TQ13 7DF

Prices do not include VAT or other local taxes, but do include delivery in the UK and Europe.
Please check prices at time of order as ads are prepared some weeks before publication.
This page lists some of our products– call us for a complete price list and details of discount structure.
ORDER BY PHONE WITH YOUR CREDIT CARD.

TEL: (0364) 53499
FAX: (0364) 53071

FORTRAN COMPILERS

NEW LAHEY F77L-EM/32 5.0 is now bundled with royalty-free version of Phar Lap's Extender & VMM, 486 optimisation, arrays > 16 Mb, new debugger, etc. ONLY £660.

FS-Fortran 77	DOS	£35
Lahey F77L 5.0	DOS	£340
Lahey Personal Fortran	DOS	£75
LPI RM/FORTRAN 3.1	DOS	£535
MS Fortran 5.1	OS2&Win&DOS	£205
Prospero Fortran	DOS	£145
Prospero PC Fortran	DOS	£55
Watcom F77 8.5	OS2&Win&DOS	£275
FTN77/386	386	£765
FTN77/486	386	£895
Lahey F77L-EM/32 5.0 & PL386	386	£660
Watcom F77/386 8.5	Win&386	£450

LOW PRICES FOR MICROSOFT & BORLAND PRODUCTS

Microsoft Basic PDS 7.1	**£222**
Microsoft C/C++ 7.0	**£222**
Microsoft C/C++ Upgrade	**£99**
Microsoft Cobol 4.5	**£435**
Microsoft Fortran 5.1	**£205**
Microsoft Macro Assembler 6.0	**£75**
Microsoft Pascal 4.0	**£139**
Microsoft QuickBASIC 4.5	**£50**
Microsoft QuickC 2.5	**£50**
Microsoft QuickC & Assembler	**£83**
Microsoft QuickC for Windows	**£95**
Microsoft QuickPascal 1.0	**£50**
Microsoft Source Profiler	**£45**
Microsoft Test for Windows	**£185**
Microsoft Visual Basic	**£95**
Microsoft VB Prof Toolkit	**£135**
Microsoft Windows 3.1	**£70**
MS Windows SDK 3.1 (in C 7)	**£222**
Borland C++ 3.0	**£199**
Borland C++ & AFX 3.0	**£295**
Borland C++ & AFX Upgrade	**£189**
Borland Corporate Prod Pack	**£330**
dBase IV 1.5	**£335**
ProtoGen for Borland C++	**£39**
Turbo C++ 3.0	**£50**
Turbo C++ & Turbo Vision 3.0	**£70**
Turbo C++ for Windows	**£85**
Turbo Debugger & Tools	**£89**
Turbo Pascal 6.0	**£69**
Turbo Pascal Professional 6.0	**£137**
Turbo Pascal for Windows	**£103**
Turbo Vision 3.0	**£25**
Paradox 3.5	**£335**
Paradox Engine 2.0	**£199**

WITH FULL TECHNICAL SUPPORT

C & C++ COMPILERS

NEW HIGH C/C++ 386 3.1 includes templates, global optimiser, Borland C++ compatibility options, supports Phar Lap, Ergo & Rational Extenders. ONLY £475.

NEW WATCOM C/386 9.0 includes 486 optimisations, generates PenPoint and OS/2 2.0 apps/DLLs (as well as DOS-386, Windows and AutoCAD ADS & ADI). Can be hosted on OS/2 2.0 and integrated with Workframe/2. ONLY £485.

NEW WATCOM C 9.0 DELTA PACK adds 16-bit support (DOS, Windows, OS/2 1.x) to C/386 9.0 for ONLY £99.

NEW MICROSOFT C/C++ 7 includes the full Windows 3.1 SDK (total over 10,000 pages of documentation!!). THIS IS NOW SHIPPING. The Upgrade includes the 3.1 SDK software only - the additional docs are available separately - call us for details.

Aztec C86-c	DOS	£260
Borland C++ 3.0	Win&DOS	£199
Borland C++ & AFX 3.0	Win&DOS	£295
Borland C++ & AFX U/G	Win&DOS	£189
Microsoft C/C++ 7.0	Win&DOS	£222
Microsoft C/C++ Upgrade	Win&DOS	£99
MS QuickC 2.5	DOS	£50
MS QuickC & QuickAsm	DOS	£83
MS QuickC for Windows	Win	£95
Topspeed C Prof	Win&DOS	£165
Topspeed C++ Prof	Win&DOS	£165
Turbo C++ 3.0	DOS	£50
Turbo C++ for Windows	Win	£85
Watcom C 8.5	OS2&Win&DOS	£275
Watcom C 9 Delta Pack for C/386		£99
Zortech C++	Win&DOS	£370
Zortech C++ S&E	Win&DOS	£550
High C/C++ 386 3.1	PL386	£475
Intel 386 C Code Builder	Win&386	£370
NDP 386 C	386	£545
Watcom C/386 9.0	Win&386	£485

PROGRAMMING TOOLS

Ada	**Algol**
Assemblers	**AWK**
Basic	**C**
C++	**Cobol**
Comal	**Comms**
Cross Dev	**Database**
Debuggers	**Editors**
Forth	**Fortran**
Graphics	**Icon**
Linkers/Locaters	**Lisp**
Logo	**Modula-2**
Pascal	**Prolog**
Rexx	**Smalltalk**
Snobol4	**Version Control**
Windows	**Yacc & Lex**

We stock many items for which there is no space in these advertisements.

C LIBRARIES & TOOLS

Comms

C Asynch Manager	DOS	£110
C Comms Toolkit	DOS	£89
Essential Comms	DOS	£150
Greenleaf CommLib 3.2	DOS	£175
Silvercomm C Asynch	DOS	£125

Database

Btrieve	Win,DOS	£310
C-tree Plus	DOS	£310
CodeBase 4.5	DOS	£195
D-ISAM	DOS	£275
Faircom Toolbox Prof	DOS	£645
Pro-C v2.1	DOS	£145
Pro-C v2.5	DOS	£495
Raima Data Manager	Win,DOS	£425
SoftFocus Btree/ISAM	DOS	£70
Sycero.C Network	DOS	£895

Graphics

Essential Graphics Chart	DOS	£110
Essential Graphics Kernel	DOS	£115
Essential Graphics GUI	DOS	£85
GraphiC 6.0	DOS	£260
graphics-MENU & Data Entry	DOS	£155
GSS GDT	OS2,DOS	£425
GX Graphics 2.0	DOS	£135
HALO Professional 2.0	DOS	£245
MetaWINDOW 3.7c	DOS	£155
PCX Toolkit 5.0	DOS	£135
Real-Time Graphics & M/CT	DOS	£145
TEGL Windows Toolkit II	DOS	£65

Screen

C Windows Toolkit	DOS	£59
C-Scape + Look&Feel	OS2&DOS	£370
C-Worthy	DOS	£389
Greenleaf Datawindows	DOS	£215
MEWEL Window System	DOS	£175
Object Professional for C++	DOS	£140
Panel Plus II	OS/2&Win&DOS	£260
ProtoGen	Win	£55
ProtoView	Win	£299
TCXL-DOS 6.0	DOS	£75
Vermont Views V3.0	DOS	£299
Vitamin C V4.0	DOS	£250

General & Systems Libraries

C Tools Plus/6.0	DOS	£90
CodeRunner	DOS	£140
Heap Expander 3.0	DOS	£85
Hold Everything	DOS	£130
PDK £1, £2, £3	DOS	£70
PowerSTOR	DOS	£85
Resident-C	DOS	£115
TesSeRact TSR Dev Sys 2.0	DOS	£69

Math & Scientific

C/Math Toolchest & Grafix	DOS	£45
Huge Virtual Array & NAT	DOS	£145
Mathpak 87	DOS	£85
Science & Eng Tools	DOS	£55

Tools

C-Clearly	DOS	£85
Clear+ for C	DOS	£140
CodeCheck Prof	DOS	£435
LALR Pro 4.2	DOS	£275
MKS LEX & YACC	DOS	£165
PC-Lint	OS2&DOS	£75
PCYACC Prof	DOS	£435
Source Print	DOS	£75
Tree Diagrammer	DOS	£75

DISK COPYING SERVICE

We can copy files to and from 600 disk formats including CP/M, CP/M-86, MS- DOS, PC-DOS, APPLE, SIRIUS, BBC, TORCH, APRICOT, HP-150, TRSDOS, AMSTRAD, ATARIST, MACINTOSH.

Our charge is £10.00 + disk + VAT with discounts on small quantities and disks are normally despatched within 24hrs of receipt. For more information call us.

GENERAL PROGRAMMERS UTILITIES

Source Doc/Xref/Pretty Printers

Flowcharting III	DOS	£175
Easyflow V7	DOS	£180
SourceDoc	OS2&DOS	£270
Source Print	DOS	£75
Tree Diagrammer	DOS	£75
RFFlow 2.0	Win	£69

Make/Version Control

MKS RCS	DOS	£165
MKS Make	DOS	£100
PVCS Version Manager	DOS	£390
PVCS Config Builder	DOS	£165
SMS	DOS	£445
Sourcerer's Apprentice Prof	DOS	£429
TLIB 4.12	DOS	£65

Linkers

OPTLINK	DOS	£125
OPTLINK/Windows	Win	£290
Phar Lap Linkloc	DOS	£560
Plink86 Plus	DOS	£300
RTLink/Plus 5.0	DOS	£290

BASIC FOR WINDOWS

GFA-BASIC for Windows	Win	£170
MS Visual Basic	Win	£95
Realizer	Win	£275

Visual Basic Libraries

Comms Lib (Microhelp)	Win	£95
PDQ Comm for Windows	Win	£69
Btrieve 5.1	Win	£310
CodeBase 4.5	Win	£195
MS VB Lib for SQL Server	Win	£300
ObjectTrieve/VB	Win	£250
PowerLibW	Win	£195
Raima Data Manager	Win	£425
VB/ISAM MX	Win	£59
Graphics Server for VB	Win	£85
PowerShoW	Win	£325
Microhelp Muscle	Win	£125
MS VB Prof Toolkit	Win	£135
QuickPak Pro for Windows	Win	£120
VBTools	Win	£79

DEBUGGING TOOLS

Bounds Checker	386	£220
Multiscope for DOS	DOS	£140
Multiscope for Windows	Win&DOS	£270
Multiscope for OS/2	OS/2	£315
Periscope I/512K	DOS	£490
Periscope II	DOS	£150
Periscope II-X	DOS	£130
Periscope/EM	386	£195
Quaid Analyser	DOS	£130
Soft-ICE	386	£340
Soft-ICE/W	Win-386	£340
Soft-ICE & Bounds Checker	386	£470

C & C++ WINDOWS LIBRARIES

Btrieve	Win	£310
CodeBase 4.5	Win&DOS	£195
dAnalyst Gold	Win&DOS	£230
POET	Win	£299
C++/Views	Win	£270
GUI_Master	Win	£380
ObjectGraphics for C++	Win	£159
ProtoGen	Win	£55
ProtoGen for Borland C++	Win	£39
ProtoView	Win	£299
Win++	Win	£145
Windows Control Palette	Win	£85
WindowsMaker Pro	Win	£545
Zinc (inc Source)	Win&DOS	£325

TPW V1.5

Borland has announced an intermediate upgrade for Turbo Pascal for Windows. TPW V1.5 will include support for Windows 3.1 (TrueType, OLE, Drag and Drop etc), the latest version of Resource Workshop (replacing the Whitewater Toolkit), Turbo Debugger for Windows, colour syntax highlighting and a SpeedBar. However, there will be no class browser as seen in Turbo C++ for Windows. US upgrade price is $50 and the package should appear within the next few weeks.

SQL Connect

Following the recent release of DataEase 4.5, DataEase UK has announced the availability of SQL Connect. The product comprises a range of plug-in modules for DataEase 4.5 that allow access to distributed SQL servers such as Oracle, Microsoft and Sybase SQL Servers, and DB2 (via Micro Decisionware's DB2 Gateway product). SQL Connect costs £395 per server. Contact DataEase UK on 081 5540582.

DOS fax

Delrina Technology has released DosFax PRO, a DOS-based fax package modelled around WinFax PRO. It allows users to fax documents from any DOS application by capturing the document during print time for transmission as fax. The product works with any Sendfax, Class I or Class II compatible fax modem. DosFax PRO costs £59 and is available from Delrina UK on 081 2073163.

VB Front-end

Want to put together a Windows-based front-end for 3270 and AS/400 systems? Take a look at Wall Data's Rumba Tools for Visual Basic. Rumba adds four custom controls to the VB toolbox and provides Form Builder, a tool which automatically translates host screen output into VB forms. A single-user licence for Rumba Tools costs £126. For further information contact Wall Data on 081 7581195.

Maths classes

Meijin++ Pro 2.0 is library of 115 C++ classes for the development of mathematical, statistical and queuing models. The latest version includes an exception handling mechanism, semi-persistent containers, numerical and statistical analysis tools, and continuous and discrete event simulation. The library supports DOS (Zortech, Borland and Microsoft compilers), UNIX V/386, Sun and DEC platforms. Price is $389 for the DOS version and is available from Network Integrated Services on 0101 7147550995.

3.1 - It's Here

It has been a busy month for Microsoft. First, the long awaited successor to Windows 3.0 has finally arrived. That's right, Windows 3.1 is here. Offering a range of improvements over its predecessor, 3.1 is almost guaranteed to be a huge success. Among its notable features are True Type fonts (scalable type-faces for WYSIWYG output), Object Linking and Embedding (OLE), multimedia and better network support. With several major PC manufacturers, including AST and Dell, committed to providing Windows 3.1 pre-installed, it would seem that Bill Gates has again backed a winner.

Then there's Professional Toolkit, a new addition to the Visual Basic product line which provides a number of new programming tools and controls for creating VB applications. There's a new Grid control which lets you create spreadsheet-like displays. Microsoft has also provided controls for MDI and OLE. The Bits Per Second ChartBuilder Control has been thoughtfully included with the toolkit (see *.EXE* Oct 1991 - *Windows Graphics Without Tears*). This enables VB applications to display graphs including 3D pie and bar charts. Other controls include 3D versions of the standard Check Box and Push Button, animated buttons which are capable of displaying several different bitmaps in sequence, and a rather nifty Gauge control which bears a remarkable resemblance to Borland's 'speedometer'.

Other features of the Professional Toolkit include support for multimedia, Pen computing and the Visual Basic Setup Kit which provides a set of tools for writing your own setup programs.

Windows 3.1 costs £99, although the upgrade price for existing Windows users is £45. The Professional Toolkit for Visual Basic is priced at £199, but until the end of May 1992, Microsoft is offering the toolkit to registered users of Visual Basic at a cost of £79. Phone Microsoft on 0734 270001 for details.

IBM ships OS/2 2.0

Just in time to meet its oft-quoted release date, IBM has announced the availability of its 32-bit PC operating system. OS/2 2.0 delivers three operating environments in one - DOS, Windows (applications run in standard mode) and OS/2. It will ship with over 25 'Applets' and includes the Adobe Type Manager for both Windows and PM. Borland is expected to make a PM-hosted C++ compiler available within the next few months and Watcom C V9.0 already supports the operating system. In response to criticism of hardware dependency in OS/2 1.x, IBM has established an OEM test lab in Florida. So far about 100 OEM models have passed compatibility tests.

Until 30 June the following promotional prices will apply: £40 to upgrade from OS/2 1.x, £50 for current Windows users, £65 for current DOS users and £90 if you don't have an operating system at all. The deal includes free 60-day helpline telephone support. For details ring 081 7470747.

For OS/2 developers there is an OS/2 2.0 Tools Conference to be held at the IBM South Bank Centre in London on 18-20 May 1992. The cost per delegate is £595 + VAT for the three days. Ring Natasha Warner on 0256 56144 for further information.

Borland's Brief

Borland has acquired two new programmers' tools from the US company Solution Systems: the popular (but pricey) Brief programmer's editor and the Sourcerer's Apprentice version control system. Both products will be re-packaged by Borland and integrated into its product line. Brief will continue to be sold as a standalone product, and special hooks will be built into the language IDEs to make using them with Brief easier. Also, some Brief capabilities will be included with the standard IDEs. According to Borland CEO Phillipe Kahn, a Windows version of the editor can be expected 'real soon'. Sourcerer's Apprentice, renamed as 'The Borland Source Control System', will be integrated into Borland's Language IDEs.

Borland hired Solution Systems' programmers when it bought the products, so it hopes to continue with development apace.

Windows Ada

Ada is now available for Windows thanks to R.R. Software. The Janus/Ada Windows toolkit works in conjunction with the Janus/Ada Professional Development System for MS-DOS. It includes an Ada interface to the Windows 3.0 and 3.1 API. It also comes with two additional Windows interfaces. The first is Ada-like which provides the Ada programmer with a simpler interface. The other Windows interface is a binding which is capable of converting existing text-only applications to Windows without re-compilation. The Janus/Ada Windows Toolkit is priced at $400. The Janus/Ada Professional Development system for 386 MS-DOS contains a validated Ada compiler and costs $500. For further information contact R.R. Software on 0101 608 2513133.

Borland C++ training ...

"Spock!"

"Yes Jim."

"This strange fragment, what is it?"

"It's code, Jim. C code, but not as we know it."

"Book me on a course, Scotty."

... RICHFORDS Training.
For developers, by developers, for the real world.

We're in a class of our own.

RICHFORDS C++ training for DOS & WINDOWS

CALL - 071-922 8819 for schedule & bookings

All trademarks acknowledged.

CIRCLE NO. 904

Windows tester

Microsoft Test for Windows is a graphical testing tool designed to create and run automated test scripts for Windows-based applications. It is targeted at software developers but can also be used by purchasers for acceptance testing. The recorder generates scripts in a flavour of Basic by trapping mouse and keyboard events. The Test DLL can also be called from any programming language that supports DLL access such as C, Pascal or Visual Basic. Test suites can be run unattended with the results being logged to a file for subsequent examination. Test for Windows costs £279. For further details, ring Microsoft on 0734 270000.

Rocky C

Developers using HP BASIC or Rocky Mountain Basic can now port their applications to almost any platform that supports the C programming language, thanks to HP BASIC to C Translator from Workstation Source. The Translator costs £2,500. Workstation Source is on 0442 876234.

Foxy Graphics

Bits Per Second has introduced a new version of its dGE graphics library for xBASE. dGE for FoxPro V2.0 is distributed as a PLB (Fox's version of a DLL) and Bits Per Second has indicated that the new release offers double the performance of previous versions and eliminates the overhead associated with 'memory-hungry TSR drivers'. dGE for FoxPro V2.0 cost £245. Phone Bits Per Second on 0273 727119 for details.

9K in UK

So called to associate it with international quality standard ISO9000 (equivalent to BS5750 over here), 9K is a new version control system from Forté Software. Of particluar interest to xBASE developers is the fact that 9K recognises dBASE files and can monitor changes in structure and data. The product is distributed in the UK by QBS Software (tel: 081 9944842) and is priced at £149 for a single-user DOS version. A multi-user LAN version will be available shortly priced at under £500.

Who said that?

Quantech, the company that gave your PC a human voice (see .EXE Dec 91 - Sounds Peculiar), has done it again. The latest version of SoftSpeak II+ is compatible with the SoundBlaster/ThunderBoard sound cards and can also use the WAV sound files of Windows 3.1. The software-only version of SoftSpeak is priced at £59.90 with the development version at £189.90. Speak to Quantech on 091 2280513 for more information.

Fox falls to Microsoft

Microsoft's acquisition of Ohio-based Fox Software signals the company's late entry into the dBASE language arena, and is seen by many as an attempt to prevent Borland from dominating the market. Borland share prices fell with news of the announcement. Commenting on the deal, George Fletcher MD Nantucket UK said 'it reveals that Microsoft is not as advanced as it had led people to believe with its own database product, Cirrus'. He went on to soothe Clipper users by saying that 'from Nantucket's point of view, there is no threat to our leadership in the application development tools business, since neither Fox nor Microsoft has a compiler at present for either FoxPro or Cirrus'.

As part of the 'merger', David Fulton president of Fox, will join Microsoft as the database architect of Microsoft's newly formed Database and Development Tools division.

The deal leaves Nantucket battling with the two software giants for a slice of the lucrative xBASE market which has been in such turmoil in recent months.

Embed with Borland C++

C++toPROM from Systems & Software allows Borland C++ to be used to generate ROMable applications. The product takes an .EXE file and converts it to .BIN format for loading into PROM emulators or programmers. The user has complete control over the placement of code in the target system as well as being able to use memory mapped I/O at absolute addresses.

Another option is the production of .OMF files required by ICE or by target monitors such as SoftProbe TX. The C++toPROM package includes all the code required to initialise the system hardware and segment registers.

C++toPROM is priced at £255 and is distributed in the UK by Computer Solutions (tel: 0932 352744) which also provides similar products that work with Microsoft, Watcom and MetaWare C compilers.

dBASE Custom Ctrl

Q+E Database/VB (QEVB) is a new custom control for Visual Basic which provides the programmer with a tool for building Windows database applications with VB. The Query control enables the programmer to set up such information as the database to access, the sort order and the query to perform in a similar manner to how you would alter the properties of a VB ListBox. Once the Query Control has been 'drawn' onto a VB form, it can be used to provide automatic access to a database whenever a given VB control is clicked.

QEVB also provides a Grid control which enables you to display all the records and fields in a database. This provides a neat way of adding a database browser to your applications. QEVB is compatible with dBASE and supports both dBASE III and IV indexing. It also supports file-locking for writing multi-user applications. QEVB is distributed in the UK by Contemporary Software who is offering it at £99 for a limited period. Contemporary Software is on 0273 483979.

Too .EXE For My Shirt

Here's a photograph of our Kate Adams (left) with her captain posing in a rather trendy rugby shirt. Notice that awesome designer label, an icon of good taste and impeccable style? That's right, .EXE is sponsoring the first East to West Coast rugby tour undertaken by a women's rugby team. 23 members of the Richmond Women's Rugby Club will be travelling from Boston to San Francisco, playing five games in all. Good luck Kate!

How Portable?

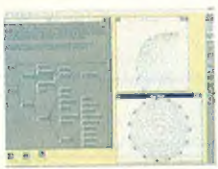
Open System Portability Checker from Knowledge Systems is a tool which allows you to determine the extent to which your UNIX applications conform to a number of standards including the X/Open Portability Guide (XPG), POSIX.1, POSIX.4, POSIX.16 and the ANSI C Standard.

Source code can be tested to ensure that it adheres strictly to a given standard. The type of objects can be checked over several modules and it is possible to determine whether function calls are given the correct number of parameters with the correct parameter-types. At run-time, Portability Checker monitors all system calls and checks the correct usage of pointers in order to track down 'out of bounds' errors. Open Systems Portability Checker is available on Sun 4 and is priced at £6,000 for a single-user licence. Knowledge Systems is on 0252 520667.

CREATE ORDER OUT OF CHAOS

With zApp You Can Spend More Time Doing What You Do Best.

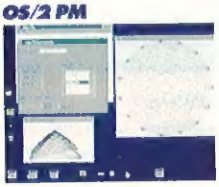
zApp is a C++ Application Framework designed to free you from the complex details of programming in Windows, OS/2 2.0, DOS and UNIX X/MOTIF. With zApp you can now cut development time and code length by up to 80%! *Leverage your development.* Why program for only one environment when you can have them all? zApp gives you single source code compatibility so you can run your applications on all the above platforms, just by recompiling. *Have all the objects you'll ever need.* zApp provides you with over 130 ready made object classes, encapsulating all the facilities you need to quickly create sophisticated applications, from Windows, Dialogs, Controls, Menus, Fonts, Graphics, Bitmaps, to MDI, Printing, Memory Management and Forms. Get up to speed (in hours, not months). zApp comes with comprehensive tutorials and over 35 sample programs, many based on examples from Petzold's "Programming Windows", and others which show you how to use facilities like 3-D tool bars and status lines. You can use zApp now with your existing Windows programs written in C or C++ and get immediate productivity gains. *Rise above the chaos and join the new order.* Try zApp risk free for 60 days. If at any time you're not satisfied, just return it for a full, unconditional refund. Call Grey Matter on (0364) 534499 or contact us via fax or CompuServe to join the thousands worldwide experiencing the benefits of zApp.



WINDOWS

DOS

OS/2 PM



UNIX X/MOTIF

SOME OF ZAPP'S ADVANCED FEATURES INCLUDE:

Simplified Window Definition Create a window in one line of code.

Unified Display Architecture (Device Context) Use the same drawing code for windows, printers, bitmaps, metafiles and OLE.

Fixed Text Pane Printf to a window without worrying about repainting.

Logical Size and Dimensioning Objects Automatically resize objects based on display metrics.

Frame/Pane Architecture Separates frame windows, with application menus, from pane windows, with display code. Pane windows may be easily attached to different frames which greatly increases code reuse and simplifies maintenance.

Automatic Window Sizing and Positioning Sizer objects easily specify how a window is resized in relation to its parent, greatly simplifying complex screen design.

Data Entry Forms Comprehensive input validation for simplified data entry dialog box creation.

Flexible Message Handling Messages can be dynamically dispatched to any object at compile time or runtime.

Transparent MDI Support Instantly convert single document applications to MDI. Menu switching and other complex details are handled automatically.

Controls Automatically superclassed at runtime, including third party custom controls.

Automatic Optimized Memory Management System Hundreds of thousands of objects can be allocated.

Advanced Printing Support Includes banding, status dialogs, and printer configuration.

Pricing

zApp for:	
Windows	**$195**
Windows w/Source	**$295**
DOS & Windows w/Source	**$495**
OS/2 & Windows w/Source	**$695**
UNIX X/MOTIF w/Source	**4thQ'92**

zApp™ C++ Application Framework

zApp requires no run time royalties, is available with source and is written in standard C++. zApp supports Borland 3.0, Microsoft 7.0, Zortech and other C++ compilers. For a free copy of the zApp demonstration software, send us a request via CompuServe or call Grey Matter. Corporate training available.

INMARK

2065 Landings DriveMountain View, CA 94043
Phone 0101 415 691 9000 Fax 0101 415 691 9099
BBS 0101 415 691 9990 CompuServe 70550 2570

Grey Matter (UK), Prigg Meadow
Ashburton, Devon TQ13 7DF
Phone 0364 53499 Fax 0364 53071

CIRCLE NO. 905

The unique windows based

Call 0276-692269

client/server application builder.

PowerBuilder™

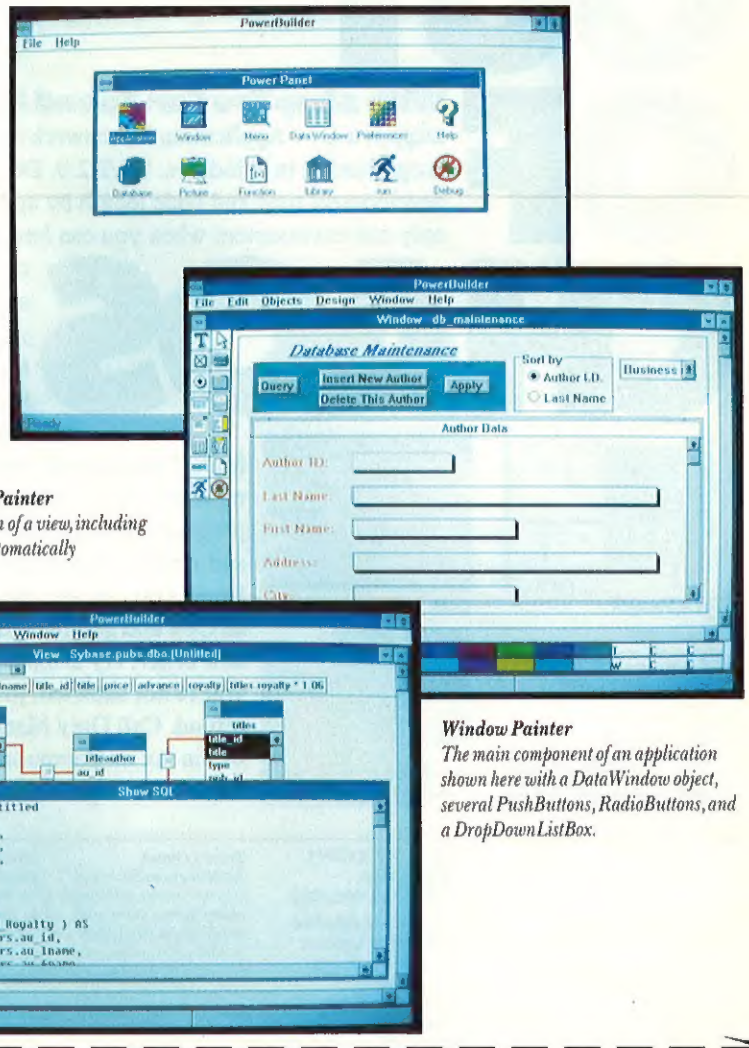
PowerBuilder is the most powerful and easy-to-use environment for building industrial-strength client/server applications. It couples an easy, intuitive graphic-user interface with your powerful server-based relational (SQL) database.

PowerBuilder takes the best characteristics of traditional systems and combines them with the best features of the new Windows-oriented, PC-based client/server architecture.

Traditional development languages like C or COBOL either don't address the client/server world or are too cumbersome for MIS developers concerned with productivity.

PowerBuilder takes a different approach. It lets you *paint* applications quickly and easily. And when your needs change, you alter them in minutes, not weeks.

Getting started is simple. PowerBuilder lets you access all of its features from one central location, called the Power Panel. Anything you need to do is just a mouse-click away.



Power Panel
This is a powerful command center for the developer, which provides easy mouse-driven access to the entire PowerBuilder environment.

Database View Painter
Graphic definition of a view, including joins, showing automatically generated SQL.

Window Painter
The main component of an application shown here with a DataWindow object, several PushButtons, RadioButtons, and a DropDownListBox.

I would like to:

- ☐ Receive a PowerBuilder Info Pack
- ☐ Attend a PowerBuilder Seminar
- ☐ Evaluate PowerBuilder

Name: _____

Job Title: _____

Company: _____

Address: _____

_____

Tel No: _____ Fax No: _____

.EXE 5/92

admiral

Admiral Software Limited
Admiral House
193-199 London Road, Camberley
Surrey GU15 3JT
Tel: (0276) 692269
Fax: (0276) 691541

Powersoft™

CIRCLE NO. 906

DOSX Born-again

Zortech's DOSX, the 32-bit DOS extender which was shipped with Zortech C++ V3.0, has been re-vamped and is now available from FlashTek under the new guise of X-32. Improvements include a debugger interface which enables developers to perform source-level debugging on the `spawn()`, `system()` and `exec()` system calls. X-32 also has a bigger brother called X-32VM which provides up to 3.5 GB of virtual memory. X-32 is compatible with XMS, VCPI and DPMI and is presently available for the Zortech compiler. FlashTek distributes X-32 together with X-32VM as a single package which costs £60 (no royalties). X-32 is available in the UK from Murray Communications on 0476 74108.

X For DOS

Quaterdeck has unveiled DESQview/X, the latest release of its popular DOS multitasker. Quaterdeck has boldly proclaimed that 'DESQview/X is the first full DOS client/server implementation of the X Windows System'.

There is support for remote computing and Quaterdeck has provided the facility to run DOS text and Microsoft Windows applications remotely on another DESQview/X PC or X workstation. It is possible to cut-and-paste between different windows irrespective of whether the applications within the windows are running locally or remotely. DESQview/X includes the Application Manager, File Manager and the Adobe Type Manager which provides 13 Type 1 fonts for X Windows and enables DOS text applications to run inside scalable windows. Quaterdeck's QEMM386 and Manifest are bundled with DEQview/X.

Support for OSF/Motif and Open Look is also available separately. DESQview/X costs £199 and comes with Network Manager which provides support for Novell Netware and NetBIOS networks.

A Network Manager for TCP/IP is also available at a cost of £149. The OSF/Motif Window Manager is priced at £189. The Open Look Window Manager costs £149. Quaterdeck can be contacted on 0245 496699.

MOT for LAN

LANprobe is a software-based network analyser which can be installed on any PC or AT machine. LANprobe supports Novell NetWare 286, Novell NetWare 386, Microsoft LAN Manager and IBM LAN Server protocols. The menu-driven software is able to capture network traffic in real time and then display the information in a variety of formats, including one which displays an annotated decoding of the protocol packets. The efficiency of the network can be displayed graphically on a PC's screen using bar graphs. It is also possible to use LANprobe to detect unauthorised accesses to the network, without alerting potential hackers.

The software is distributed with a network card which has been programmed to capture all network traffic. LANprobe costs $995 and is produced by General Software on 0101 206 3914285.

Are you Experienced?

Experience for Windows from Expert Edge adds another product to the burgeoning catalogue of Windows application development tools.

The product combines visual design tools with a 4GL-like programming language that can be used to develop database applications. File formats are compatible with dBASE data and index files. There is also support for DLL access from the scripting language and DDE support. The company is planning to provide links to a variety of SQL servers in subsequent releases.

Experience for Windows is currently available at an introductory price of £495 (a saving of £200). A run-time version is also available for £795 (saving £300). Expert Edge is on 081 9417433.

PC Teletext

Tune into the Ceefax and Oracle teletext transmissions with Optimum Technology's OPT-III/S teletext receiver. This external box-of-tricks plugs into a PC's RS232 port and enables you to view pages on-screen, or save the information to disk. By converting the raw data to ASCII, it is possible to load the information into a database or spreadsheet.

The software that comes with OPT-III/S includes a utility to set the system clock to the teletext time, sequencing software which provides a batch file mechanism for automating the process of selecting which pages to view/print, and the OPTIII.SYS device driver which provides a mechanism for controlling OPT-III/S from within your own software. OPT-III/S External Receiver is priced at £295. An internal version is also available at a cost of £195. For more information phone Optimum Technology Ltd on 081 2030220.

Objects of '92

Object Expo-Europe is a new annual exhibition of object-oriented technology, which will take place on 14-17 July 1992 at the Church House Conference centre, Westminster, London. There will be over 40 companies exhibiting OOP products and more than 50 lectures, tutorials and panel discussions. Details of the conference can be obtained from David Lloyd on 071 306 77331.

From C To C++

Bjarne Stroustrup, the father of C++, will be speaking at the C Plus C++ In Action conference at the London Gatwick Airport Hilton on the 8-12 June 1992 in which he will discuss the transition from C to C++. 'In Europe, I have more problems explaining that C++ doesn't have all of C's faults', said Stroustrup, '...even C doesn't have all the faults ascribed to it!' Details of the conference can be obtained from the Boston University Conference Office on 071 2592032.

3 Day GUI

Cambridge Connectivity promise to give one of your character-based UNIX applications a GUI in 3 days or your money back. Using the Soft Option development kit one of the company's engineers will spend 3 days on-site and knock up a windowed version all for £3,950 which includes a copy of the kit (extra copies priced at £2,000). For more information on Soft Option and the current promotion contact Cambridge Connectivity on 0954 51968.

3270 Emulator

Select Windows Client V1.1 is the latest release of Digital Communications Associates' 3270 terminal emulation package for Windows. Select Windows provides up to five 3270 sessions and 64 LU 6.2 sessions. Each session can be displayed in a resizeable window. DCA Select Windows Client V1.1 costs £2,749 for a comms server-based licence. DCA can be contacted on 0442 231414.

Open Look Debug

TauMetric's C++ debugger is an Open Look-hosted debugger for the TauMetric C++ compiler on the Sun SPARCstation. Program information including source code, program I/O and scope information is displayed in one of five windows. There is also a graphical class browser which is able to display instant variables using a point-and click method to dereference pointers. TauMetric's C++ debugger and browser costs £1,200 and is distributed in the UK by Instrumatic Ltd on 0628 476741.

Letters

We welcome short letters on any subject that is relevant to software development. Please write to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. Unless your letter is marked 'Not for Publication', it will be considered for inclusion in this section.

Beer money

Sir,

I am compelled to write to you after a recent software supply experience. The prospective client was a 'computer literate' hotel owner. The client had purchased an integrated accounts package in the dim hope that it would satisfy his requirements. The package was eminently unsuitable for his purposes. This left him wary of software suppliers. In an effort to allay his fears, I produced a system outline documentation and was greeted with, '...anybody could run that off on a word-processor'. Temper in check, I patiently explained that this document provided the basis for an agreement between ourselves which would be referred to and amended as the implementation progressed. Unimpressed, the client changed tack and stated that he could probably get some newly qualified computer person to do the whole thing for 'beer money'.

The point I'm trying to make is that there isn't enough control over software providers and that 'computer literate' is too ambiguous a term. I am a conscientious developer and will only supply software that I know will do the job. I'd rather lose a client than supply to make the sale (poverty doesn't frighten me) - I'll probably lose this one when the hotel owner receives the quotation for a bespoke system I've sent!

Roland Lees
MicroSolus, Motherwell

Too Extreme

Sir,

Regarding the article *Are hackers really criminals?* by David Martin in *.EXE* April 1992. While the factual content of the article was, well, factual, and such an article was certainly necessary, I think if the ranting opinions and vitriol are stripped out of the piece one is left with perhaps only half the number of words.

While Mr Martin is certainly entitled to his opinions, his assumption that anyone who does not subscribe to his views is ignorant or stupid is puerile. His belief that hacking will be stopped by this legislation is simply wrong. Security will still be required, for the same reasons as before.

Mr Martin believes that hacking can and should be eliminated by ever more vicious penalties. In no other part of our law is anything equating to trespass even a criminal offence, but thanks to the CMA it carries a scale five fine. How does he justify this? He then criticises Judge Francis Aglionby for not understanding the reasoning behind the law. Good God, man, I don't understand the reasoning (and neither, if truth be told, does he)!

The CMA, like the DPA, is useless. They were drafted as a response to public worries about banking security, credit references, and the like; being thrown together by other uninformed people they do not address the real issues. Mr Martin, you claim to be an expert. You have a responsibility to inform in a rational manner, rather than to propagate the kind of idiocy and scaremongering that brought the acts into existence in the first place.

Jules May
Herts

Increased privacy

Sir,

Peter Sabine-Bacon's Soapbox article in March's .EXE was in my opinion very much along the right lines. In the final paragraph he states that the technique he proposes does not hide `private` member functions. The solution to this is very simple.

Rather than place the `private` members of a class in a `struct` as he proposes, place them in another class where the `private` member functions can also be declared. His example then becomes:

Public Header File (for distribution)

```
class ObjectImp;
class Object
{
private:
  ObjectImp * pImp;
public:
  Object(int data);
  ~Object();
  int GetData();
};
```

Private Header File (not for distribution)

```
class ObjectImp
{
friend class Object;
private:
  int Value;
  int PrivateFunc();
};
```

Source File

```
#include "<public header>"
#include "<private header>"
Object::Object( int data )
{
  pImp = new ObjectImp;
  pImp->Value = data;
}
Object::~Object()
{
  delete pImp;
}
int Object::GetData()
{
  return pImp->Value;
}
```

I don't claim this idea as mine originally, as I first saw it in the header files for Glockenspiel's CommonView.

Nicholas King
Olivetti Systems & Networks

Paul Kemp replies - actually I wrote the article based on work with Peter. Your proposed extension is indeed an improvement, and I think is a good model for suppliers of C++ libraries and developers alike.

Letter of the Month

The writer of the best letter of the month, as judged by the Editor, will receive a £20 book voucher, courtesy of Just Computer Books. The best letter is the one printed first. Please note that letters submitted to this page may be edited.

Put an end to software piracy!

Meet the growing family of security keys from Software Security.

Each one a specialist at enforcing your license agreement in virtually any user environment you can think of. Whether it's DOS, UNIX, Macintosh or OS/2... whether it's a single user installation or a LAN.

Simply connect the appropriate key to a single user computer, or a non-dedicated file server in a network, and you control all access to your protected application.

Users, however, won't even know it's there. The keys are transparent and won't impact software functionality or the ability to make back up copies. Normal node and LAN operations are unaffected.

Simple. Unassuming. Ever vigilant. Easy to incorporate into your application package. And quite possibly the most profitable hardware investment a software developer can make.

To find out more, call:

(0784) 430060

THE ULTIMATE PROTECTION FOR SOFTWARE PUBLISHERS

Software Security International Ltd

21a The Precinct, High Street, Egham, Surrey, United Kingdom, TW20 9HN

National telephone: Egham (0784) 430060 fax: (0784) 430050 International telephone: +44 784 430060 fax: +44 784 430050

An introduction to OS/2 2.0

OS/2 version 2 is IBM's first truly integrated operating platform for DOS, Windows and OS/2. Michael Price takes a first look for .EXE Magazine.

OS/2 2.0 is IBM's first version of OS/2 to support 32-bit, finally exploiting the full capabilities of the Intel 386 and 486 processors. The flat memory model allows programs to address memory as one single huge address space, and demand paging makes it feasible. However, OS/2 2.0 has not forgotten its origins. It can still run 16-bit OS/2 applications without change. Windows 3 binary compatibility is provided in a protected environment. DOS applications are supported using the virtual 8086 mode, to provide multiple virtual DOS machines. All processes run in protect mode, and the virtualisation of interrupts and hardware access allows even communications programs to run in the background.

OS/2 2.0 still offers full multi-tasking, multi-threaded pre-emptive operation and installable file system support, with HPFS (High Performance File System, an alternative to the FAT-based file system), CD-ROM support and an enhanced version of the FAT file system. And the OS/2 development tool kit has been upgraded and is joined by some new products to help take advantage of the simpler 32-bit programming model. The observations in this article are based on the interim release 6.304E of OS/2 version 2.

Installing OS/2 2.0

When you acquire a new operating system for a machine, the first down-to-earth practical consideration is fixed disk organisation. The simplest approach is to install OS/2 2.0 as the main operating system on a single partitioned disk. This is the simplest approach and is used when OS/2 2.0 will be used all the time. It does give the option to boot DOS, either through dual boot (with FAT formatting) or as a 'guest' system.

OS/2 2.0 also supplies a boot manager facility which gives up to three alternative primary partitions, each with a separate operating system (eg MS-DOS 5.0, OS/2 1.x and OS/2 2.0). Of course, only one of these partitions can be active at any one time, so files cannot be shared between the separate systems. The boot manager needs 1 MB of disk space at the start or at the end of the fixed disk. As long as there are no more than three primary partitions, the remaining disk space can be divided into one or more logical partitions that stay accessible. Each operating system can access the files in the logical partitions, so it is possible to share, for example, the SWAPPER.DAT file between OS/2 1.x and OS/2 2.0.

As well as the many partitioning options, OS/2 2.0 is equally generous in its options for actually installing the code. As usual, the product can be installed from diskette, either 3.5" or 5.25". The installation must be from the boot floppy drive. The improved installation process appears to work faster, and effective use of graphics illustrates the progress of the installation. However, with twenty diskettes for the base version, and more again for extended services, this can be a long and tedious process.

The installation is heavily tailorable. A minimum system takes less than 20Mb of disk space, while selecting all the optional features, including reference books and additional fonts, takes in excess of 40Mb of disk space. OS/2 2.0 offers some pre-defined selections, or a free choice, and it is possible to go back later and add any features that may have been missed the first time round.

OS/2 2.0 includes a migration database that includes specifications of various DOS, Windows and OS/2 programs. This enables the system to search for all matching applications on any of the accessible drives, and place them in folders on the OS/2 desktop, with the appropriate settings. A custom version of the database file can be created, to ensure that required applications get added automatically.

The Desktop

The appearance of OS/2 2.0 is different from any of the previous PC operating environments. In the workplace shell, everything is on one screen, the desktop (Figure 1 - Desktop and Folders). There are four types of object associated with the desktop - data file, program, device and folder. The folder is a container for other objects, and the desktop is itself a special folder that fills the entire screen.

When OS/2 2.0 is installed, the desktop is created with some initial folders, including OS/2 System, Information and Templates. Additional folders will also be created with migration of existing Windows or DOS applications, or installation of additional components such as the extended services (for database or communications functions).

Each object has an associated pop-up menu, displayed using mouse button 2. This offers copy, move or delete functions for the object. Opening the object gives access to an additional menu through which the characteristics of the object can be set or changed. Double clicking with mouse button 1 starts program objects. It is also possible to associate programs with specific data file types (for example, .DOC files could be associated with DW5/2) and then double clicking on the data object will cause the appropriate program to be started. By default, data file objects will be associated with the system editor.

This new way of working may be too much of a change for some users who are used to a particular environment. For these, OS/2 2.0 offers the option to make the system look like OS/2 1.x or Windows. It is necessary to boot the system from an OS/2 diskette (typically Install diskette #1) in order to modify the OS2.INI file. The modification is done by running the MAKEINI program from the \OS2 directory; the command

```
MAKEINI OS2.INI OS2_13.RC
```

produces the appearance of OS/2 1.x when the machine is next booted from fixed disk, while

```
MAKEINI OS2.INI WIN_30.RC
```

starts a Windows 3 emulation.

These procedures provide the appearance of the selected operating environment, but not the functionality. They will however make it easier for users to find the facilities they require.

OS/2 applications

Normally OS/2 applications will be installed as program objects on the desktop or in appropriate folders and started using the mouse buttons or the drop-down menus. However, OS/2 applications may also be started from an OS/2 command line.

The OS/2 System folder contains a Command Prompts folder which, in turn, contains icons for full screen or window OS/2 sessions. When one of these is selected, the appropriate session is started, permitting applications to be initiated from the OS/2 command line provided.

Re-selecting the OS/2 prompt icon (which will be shaded to show it is in use) will merely switch the focus to that session. To start additional OS/2 sessions, copy the icon into the same folder or a different folder, and then start that new icon. There can be many OS/2 sessions, subject to the limits of the available memory (although the new paging approach to virtual memory allows a much higher level of overcommitment than previous versions).

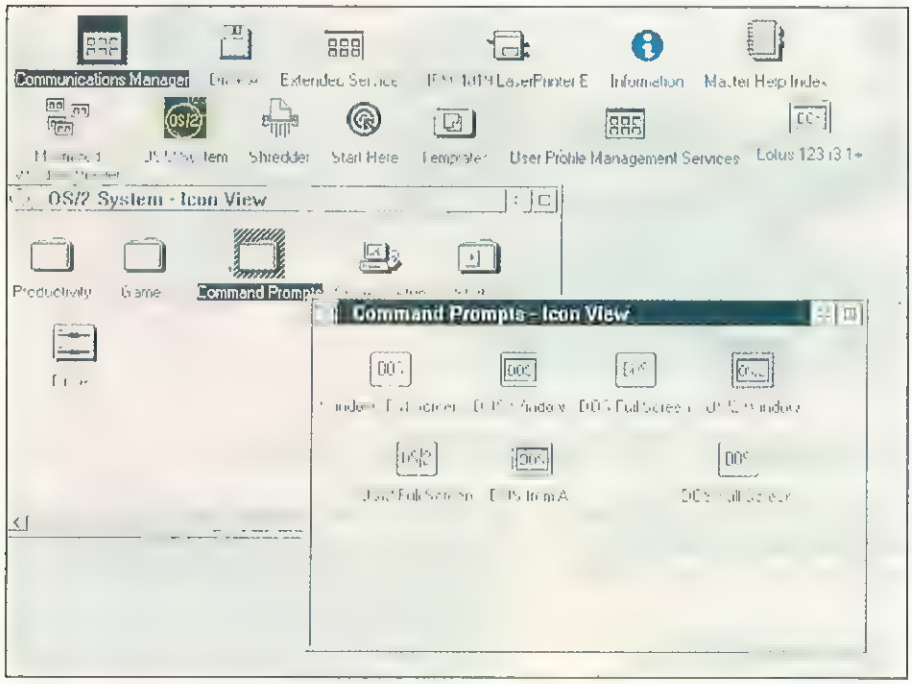


Figure 1 - Desktop and Folders

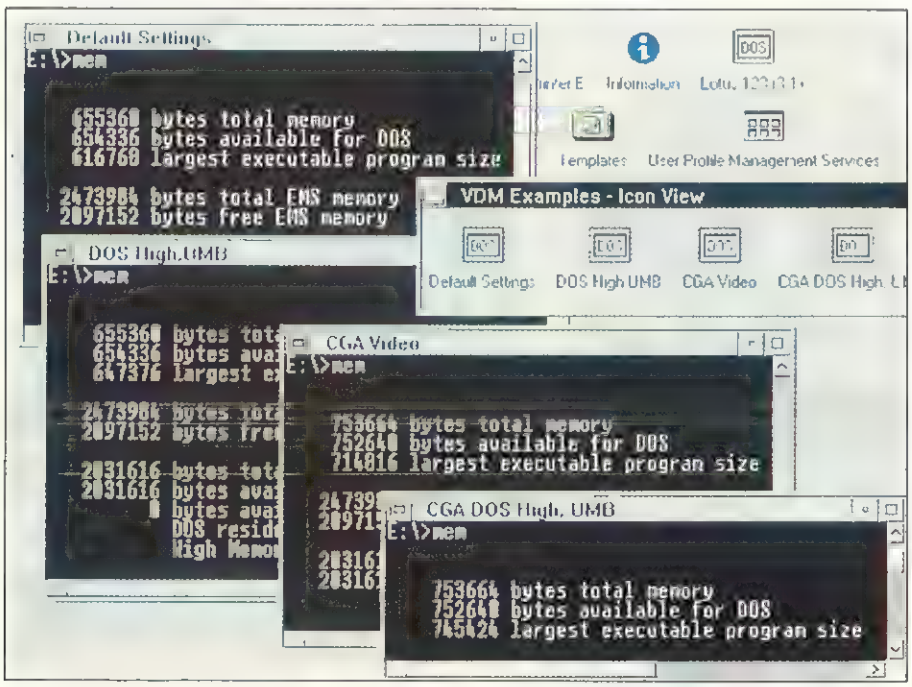


Figure 2 - Virtual DOS Machines

Running DOS

The Command Prompts folder also contains icons for full screen or window DOS sessions. These are virtual DOS machine (VDM) sessions which utilise the virtual 8086 mode of the Intel 386 and 486 processors. There can be up to 240 DOS sessions defined. As with the OS/2 sessions, additional VDMs are accessed by duplicating the icons. Alternatively, to start applications from the desktop or a folder, you can associate program objects with DOS full screen or window sessions

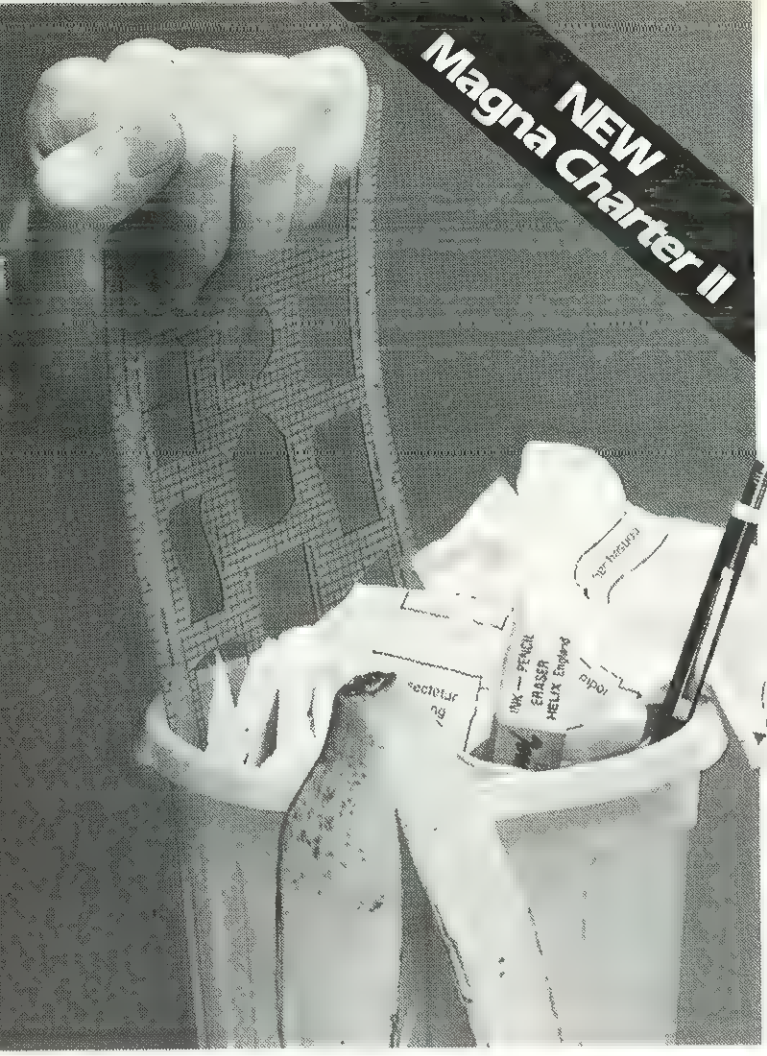
The VDMs can be extensively tailored to suit the requirements of particular applications. There are settings for the keyboard, mouse, and printer, and a range of settings for memory, including DPMI, DOS high, DOS size, UMB, EMS and XMS. Controls for video include mode, memory allocation, retrace, ROM mapping and window refresh. Other settings of interest are the startup drive, version data, and background execution.

These DOS sessions bear little resemblance to the OS/2 DOS box of old. As well as having multiple sessions, OS/2 2.0 relieves the severe memory constraints. With the default settings, VDM sessions will make 602 KB out of 640 KB available. With settings of DOS high and UMB this rises to 632 KB, with over 40 KB of free UMB space also available (see Figure 4 - Virtual DOS Machines). Moreover, all this space will normally stay available for programs, whatever device drivers are needed, since the devices will actually be supported by OS/2 device drivers linked to virtual DOS device drivers.

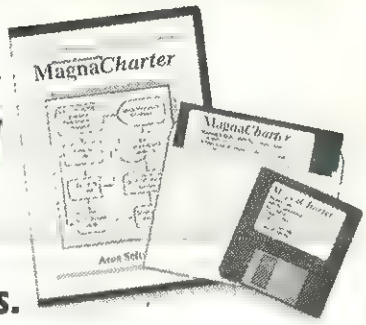
If this is still not enough, restricting the video mode to CGA will allow a DOS session of 736 KB, of which 698 KB is available for applications. With CGA, DOS high and UMB, programs can be as large as 728 KB.

Windows

Windows applications can be started from an OS/2 session merely by typing the application name. For example, typing WINWORD will launch Word for Windows in a WIN-OS2 session. Kernel support for Windows is included in OS/2 2.0 and it can run existing Windows 2.x and Windows 3 (standard mode) applications unchanged. Windows 3 Enhanced mode is not supported; IBM says this is not a problem, because the extra facilities (extra memory, multiple DOS sessions etc) offered by that mode are duplicated by OS/2 itself. Pro-



Out with the Old In with the New



MagnaCharter II relieves the pain of drawing charts.

MagnaCharter was one of the success stories of 1990. **MagnaCharter II** is the latest version of this much acclaimed product — for Microsoft Windows 3. Based on the spreadsheet principle, **MagnaCharter II** lets you build any kind of chart in minutes rather than hours and includes all the standard flowcharting symbols. Use the "crowsfeet" for database diagrams.

MagnaCharter II divides the screen into cells, identified by a double mouse click or keyboard entry into which symbols are placed. All available symbols are displayed as icons — or add your own. The various features are accessed via drop down menus, dialogue boxes and multiple windows with a wide range of text styles and sizes available. Editing is by simple cut-and-paste techniques and links between symbols can be "auto" or "conditional". **MagnaCharter II** supports PostScript output and a wide range of printers. Through the Clipboard, pass bitmaps, metafiles and text.

MagnaCharter II supports Windows 3 (**MagnaCharter** 1.03 is still available for DOS). Published by Systemstar, **MagnaCharter II** is available for £160 excluding VAT and P&P. Price includes a complete manual and examples to help you achieve professional results fast.

Order yours today by contacting Systemstar on (0992) 500919.

SYSTEMSTAR LIMITED

1-3 Parliament Square, Hertford, SG14 1EX
Telephone: (0992) 500919 Facsimile: (0992) 554261

CIRCLE NO. 909

C BEYOND 640K DOS

Let Phar Lap's 286|DOS Extender™ turn your Microsoft C compiler into a multi-megabyte power tool.

It has never been so easy to C beyond 640K DOS. With your copy of Microsoft C and the Phar Lap 286|DOS Extender you have all the tools you need to build multi-megabyte, protected mode applications quickly and easily — often by simply relinking without making source code changes. 286|DOS Extender enables you to build programs that have room for more features and capabilities without having to suffer with overlays or EMS. Best of all, you do not have to forsake your Microsoft C tools, including your Code View debugger. **No more "out of Heap" messages when compiling under Windows or with networks.**

Total Compatibility

Because 286|DOS Extender is embedded into your program, it is invisible to the end-user. Your program looks exactly like any other DOS application. There is no new operating environment for your end-user to buy or learn. Any of the 30 million 80286, 386 or 486 PCs that can run DOS can run with 286|DOS Extender. Since Phar Lap products support the XMS, VCPI and DPMI standards, applications built with 286|DOS Extender can also run with DESQview and all modes of Microsoft Windows.

So if DOS is looking smaller than ever, call Systemstar today on (0992) 500919 and C what it looks like beyond 640K — we open a world of memory.

SYSTEMSTAR LIMITED

1-3 Parliament Square, Hertford, SG14 1EX
Telephone: (0992) 500919 Facsimile: (0992) 554261

CIRCLE NO. 910

gram objects for Windows applications can also be defined, as for DOS and OS/2 applications. Windows programs may be run in full screen sessions, or in their own OS/2 desktop windows.

There are some graphics mode restrictions running DOS and Windows applications. Only full screen Windows sessions are supported with an XGA adapter with the desktop installed as XGA. Some applications may also be suspended when in the background. WIN-OS/2 does run on systems with VGA adapters, or with XGA adapters where the desktop is installed as VGA.

DOS Versions

There may be need for a specific DOS version to run a program that does not run correctly in DOS emulation. For example, if a DOS program uses a device whose driver does not load into an emulated DOS session, a real DOS session will be required.

The first step is to create a method for starting the specific DOS version. This can be a bootable diskette, a DOS partition on drive C, or an image file on fixed disk.

The path and file name of the specific DOS must be entered in the DOS startup drive setting. The FSFILTER.SYS must be included in the CONFIG.SYS to give access to the OS/2 file systems. These requirements are described in detail in the general help facility provided on the desktop. This approach will load any kernel that would run on the 8086 processor, including, for example, MS-DOS 5.0 or DR DOS 6.0.

Programming Tools

The IBM OS/2 2.0 developer's toolkit provides a comprehensive selection of language-independent build tools, productivity tools and sample programs, on-line reference information and a kernel debugger. This toolkit can be used with the IBM WorkFrame/2, which is a configurable project-oriented application development environment featuring an SAA/CUA conforming user interface. It is built with an open interface to serve as the integration point for the tools in the developer's toolkit as well as tools supporting C and other languages.

IBM continues its move away from the Microsoft camp with C Set/2, a 32-bit SAA C compiler with run-time libraries, which generates native code for OS/2 V2.0. It also includes a fully interactive, full function, source level PM debugger. For 16-bit developments, existing compilers and tools can of course continue to be used. WorkFrame/2 allows both 16 and 32-bit OS/2 tools to be plugged in.

There will also be an OS/2 2.0 version of the System Performance Monitor/2, an integrated package of performance monitoring and analysing facilities. SPM/2 enables developers to monitor and verify performance objectives, to analyse problems and as an aid for tuning and load balancing. SPM/2 can be run on a stand-alone machine, when data is usually captured for later analysis. It can also operate across a LAN, and in this case the captured data can be displayed in real time on a separate machine.

With CGA, DOS high and UMB, programs can be as large as 728 KB

Applications design

OS/2 2.0 takes full advantage of the 386 processor flat memory model and paged virtual memory. This means that applications do not have to manipulate selectors and offsets, but can view memory as a large, linear address space addressable by 32-bit offsets from the start of memory.

OS/2 2.0 provides different names and entry points for all 16-bit and 32-bit APIs, which makes it possible to mix 16-bit and 32-bit code and calls within a single EXE module. Only 16-bit or only 32-bit APIs can be used in a single C module however, because of include file support. The same type names are used in both environments. For example, take a statement such as:

```
LAB buffer = "hello";
```

In the 16-bit environment this is expanded to:

```
char far *buffer = "hello";
```

but in the flat model 32-bit environment, where everything is addressable, this becomes:

```
char *buffer = "hello";
```

The 32-bit memory layout of OS/2 2.0 is accomplished on the 386 by creating a very large segment, which can be up to 4 GB in size. A code segment and a data segment are mapped onto this segment, and `near` addresses are used throughout. Two flat selectors are used, and the base addresses of these are always zero, giving 0:32 referencing.

To make things compatible with 16-bit applications, tiling is used to map the area into 64 KB regions. The maximum address in the tiled memory region is 512 MB. The addresses are referenced using 16:16 addresses. The first 512 MB of the OS/2 2.0 virtual address space forms the compatibility region.

Technical library

The new version of the OS/2 operating system is copiously documented. There is an *Application Design Guide*, three volumes of the *Programming Guide*, three *Device Driver Development* publications, two REXX manuals and two SAA CUA manuals. Available in both hard copy and 'on-line' form are the *Control Programming* reference, three volumes of *Presentation Manager* reference, the *System Object* manual and the *IPF* manual.

These manuals are aimed at the professional programmer, and they deal with migrations from 16-bit to 32-bit as well as from DOS or Windows. All this material is additional to the manuals provided with OS/2 2.0 and associated products.

For the independent software developer, further help is available from IBM's developer assistance programme. This offers a range of facilities including OS/2 products and publications on CD-ROM, an electronic bulletin board service, and specialist workshops to aid in porting and testing 32-bit applications. This programme is run from the UK and provides cross-Europe support.

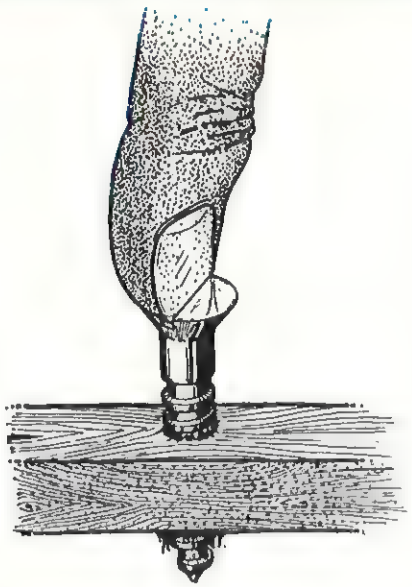
Summary

This has covered just some of the features of the new version, but it may begin to show that OS/2 2.0 is not just for OS/2 developers. DOS and Windows applications can run just as well in this new environment. When building new applications, the multiple, large memory, protected mode and independent sessions could be a positive boost to development productivity, whatever the target.

EXE

Michael has worked in the mainframe environment for many years, in technical and development roles, and switched in the early eighties to the architecture and design of systems based on PCs and local area networks. Currently, Michael is a systems design consultant in the financial industry arena.

8051 Programming?



If you've tried this, you'll know what it's like using 8051 assembler.

Admittedly, given enough time and effort, even the impossible can be accomplished.

But can your project afford to wait?

By opting for **Keil C51**, you will get better results faster and without pain.

C51 is ***very*** simple to use and yet yields the fastest and smallest code of any 8051 C compiler. With direct access to the 8051 cpu at a bit/register level in C, assembler programmers feel instantly at home.

C51 Professional Edition Comprises:

* *Full ANSI C implementation for the 8051 family*
* *Near-assembler code efficiency*
* *Function-level optimisation*
* *Extensions for 80C517/87C751*
* *Small/Compact/Large/Banked memory support*
* ***DScope51+** cpu simulator/debugger*
* ***A51** macro assembler*
* *8051 Real Time Executive*
* *Support from official C51 User Group*

To see how **C51** takes the pain out of 8051 programming, call now for our comprehensive information pack.

Hitex (UK) Ltd.
Warwick University
Science Park
Coventry, CV4 7EZ.

Tel 0203 692066, FAX 0203 692131

KEIL ELEKTRONIK

CIRCLE NO. 911

Blast! The 640K DOS barrier . . .

Let's face it. Today's average PC sports at least 1 meg of RAM, often more. So its natural that your programs should be able to make the most of the many benefits of Extended Memory. Bring on the dynamite – the TopSpeed DOS Extender Toolkit. Not only does it give you an extended development environment, it also gives you a powerful toolkit to produce DOS programs that can make full use of extended memory.

- Provides 512 Mb virtual address space
- Full range of DOS calls.
- OS/2 family API
- Supports XMS, DPMI, and VCPI
- Comprehensive porting guide
- Multi-threading and DLL support
- Supports C++, C, Pascal and Modula-2
- Royalty free distribution

For a detailed information pack on the TopSpeed DOS Extender and other TopSpeed languages call us now on 0234 267500.

Much existing code can be ported to the TopSpeed DOS Extended Environment without re-coding. The TopSpeed DOS Extender Toolkit supports TopSpeed C++, C, Modula-2 and Pascal and requires a 286 processor or better.

TopSpeed products are available in the UK from Grey Matter 0364–53499, System Science 071-833-1022, The Software Construction Company 0763-244114 and other leading software tool vendors.

JPI, The Mansards, Tavistock Street,
Bedford, MK40 2RX
Tel: 0234-267500. Fax: 0234-217094

* Special pricing for existing users

Blast the 640K DOS barrier with TopSpeed

CIRCLE NO. 912

Positioning OS/2 2.0

OS/2 1.x failed because of 286 code. Windows 3.0, the current favourite, is technically unsound; it must be superseded. Martin Healey wonders which will inherit the mantle: OS/2 2.0 or Windows NT?

The original concept of a PC for individual personal computing is still valid, particularly with portables, but the corporate world is now seeking to integrate users. This is done by interconnecting PCs using LAN technology, the PC acting as a workstation (the client). Shared resources, in particular files, databases, print spoolers and communication gateways reside on one or more servers. PC hardware can also be used for the server but this role will soon be transferred to RISC-based hardware, which is far more cost-effective for larger servers.

The essential difference between a personal computer and a workstation is that the latter must include communications functions with its operating system, while communications is an application on a personal system. The connectivity services of the workstation software should be transparent to both user and programmer and should be robust (protected mode). Thus a 386 PC with MS-DOS is a personal computer while the same hardware with OS/2 or UNIX is a workstation by the above definition.

MS-DOS versus the Mac

The inadequacy of the character-based user interface of MS-DOS has been sharply brought into focus by the Apple Macintosh. The Mac provides a consistent environment for applications, while all DOS programs are different, resulting in excessive training and support costs. Furthermore, multi-session working is a natural enhancement to any user interface (virtual screens on CDOS or UNIX), which can be enhanced if independent sessions can be activated in graphical windows. This was the advancement offered by Mac Multifinder and on the PCs by Windows 3 or DESQview (in contrast to Windows, Windows 2 and GEM which only provided multiple windows to the one program). Once multi-session, windowed interfacing is established, the well known enhancements such as cut-and-paste, drag-and-drop and icon driven commands become particularly effective.

Unfortunately, the combination of transparent communications services and multi-session working requires a rather sophisticated operating system, preferably one with true pre-emptive scheduling and multi-tasking. The primary examples of this were the enhanced UNIX executives developed for engineering workstations, eg SunOS, HP/UX or AIX. The engineering workstations and the Mac never suffered from MS-DOS, the epitome of single-tasking software, and hence provide excellent workstation/network solutions.

Because of the existence of MS-DOS and MS-DOS applications, there have been many brave attempts to create networked systems which provide some continuity. These fall into two classes: a) DOS workstations with a shared file server, dominated by Novell Netware and b) multi-session, windowed executives on top of DOS, dominated by Microsoft Windows 3, but better implemented by DESQview. The sooner it is recognised that mounting executives on top of DOS, of any sort, is only a short term solution, the better. These products are expedient at best; they partially solve the user interfacing and networking problems but, because of the lack of protected mode software, they create even bigger problems with support and training.

Continuity

Having invested in DOS applications, users are loath to throw them away. So the full

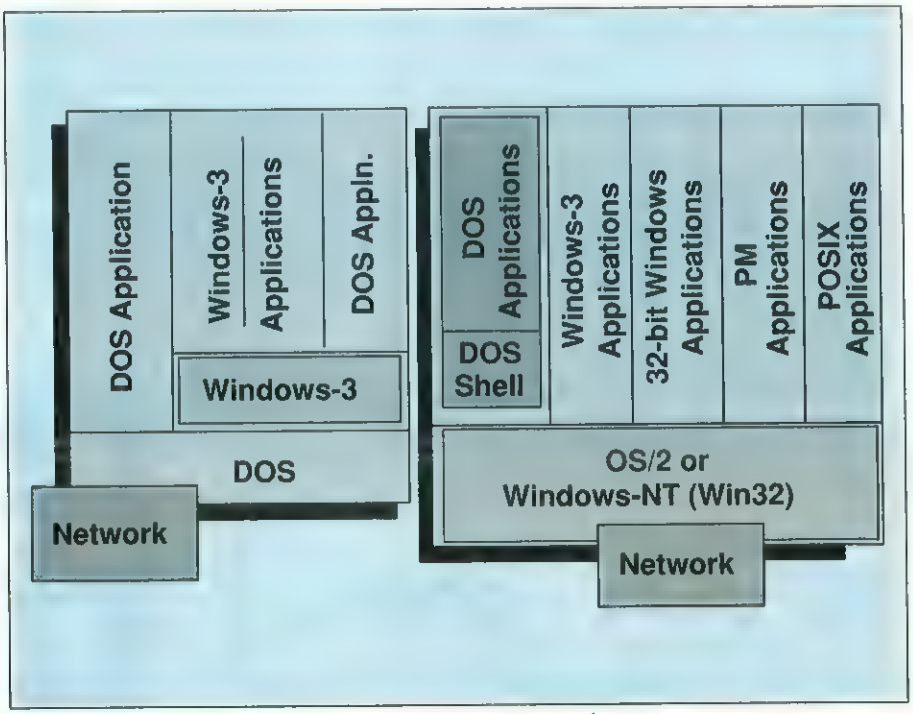


Figure 1 - Workstation OS Architectures

LEARN THE POWER OF FOXPRO 2.0 IN 5 DAYS FLAT

FoxPro 2.0 is here - and with it, the ability to process information at unheard of speeds.

Now you can see how to design, build and manage systems which do full justice to the power of the new product - with F1 training courses.

As beta-testers for FoxPro 2.0 - and a major developer of commercial database applications - no-one is better qualified to train you than F1.

5 DAY INTENSIVE COURSE

A single course which trains you in all elements of FoxPro 2.0 - five days later you can design and develop a multi-user system from start to finish.

Includes: • Database design • Program design • Program implementation • Debugging • Project management • Documentation • Reports • Screens • Menus • Mouse and window object-oriented programming • Use of Rushmore • Performance tuning • Network problem solving... and more.

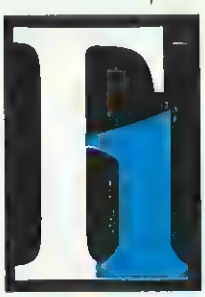
COURSES FROM 1 - 4 DAYS

F1 tailors other database courses to suit the needs of specific users. These courses range from one to four days, and provide a complete skills package.

• *For New Users:* 2 one-day courses • *For Intermediate Users:* 2 day course • *For Advanced Users:* 4 one-day courses. A detailed breakdown of all these courses is available from F1.

BOOK NOW

All courses are held at F1's training facilities in Bath or London, working in small groups of 4-6, for the highest standards of personal tuition. Call Kevin Edwards now on **(0225) 427285**, for course dates and booking information.



CIRCLE NO. 913

"Enough to make you a LATE DEVELOPER"... PC Business World July 91

Imagine a powerful integrated development environment common to whichever languages you choose, capable of supporting DOS, Windows or OS/2 development embodying the latest in OOP technology – it's called ***TopSpeed*** from JPI. Its modular architecture opens up new realms of choice–simply slot-in languages, source libraries, and toolkits to build a development system that meets your needs. No redundant components – what you want is what you get!

***TopSpeed* Environment** – Multi-window editor, powerful Project system, Hypertext help (environment, all languages and library), debugger, syntax checker, menu or command line driven, EMS support, pop-up calc, plus . . . plus . . . DOS or OS/2 **£59**

***TopSpeed* TechKit** – For power programming. Supports DOS DLLs, post mortem debugging, advanced overlay manager, TopSpeed assembler, .EXE file compressor, .OBJ file disassembler. **Windows 3 resource compiler**, Windows 3 run-time libraries, plus . . . plus . . . **£59**

***TopSpeed* C++** – makes C++ lean and mean Unique to JPI – TopSpeed is the only true AT&T 2.1 C++. TopSpeed C++ gives you SmartMethod® Linking which eliminates unreferenced classes, methods and even virtual methods – a real breakthrough for OOP programming. Includes short based pointers, concurrent tasking – even with DOS, plus . . . plus . . . DOS or OS/2 **£59**

***TopSpeed* C** – *the standard is enhanced.* The only ANSI certified C. Generates compact high quality code for DOS, OS/ 2 and Windows 3. Includes run-time error checking, multi-threading, mixed memory models, links to C++, Modula-2 and Pascal, plus . . . DOS or OS/2 **£59**

***TopSpeed* Modula-2** – the world leading implementation of Modula-2. This strongly typed and highly structured language includes type safe conversion between objects, OOP extensions with true multiple inheritance, virtual pointers, based pointers, links to C, C++ and Pascal, plus . . . DOS or OS/2 **£59**

***TopSpeed* Pascal** – the next generation. Power-up your Pascal, convert to TopSpeed, then tune it up with the hottest compiler for DOS, Extended Dos, OS/2, or Windows. ISO 7185 conformant, Turbo to TopSpeed converter, ISO conformant arrays, dynamic strings, separate compilation units, OOP extensions plus much more . . . DOS or OS/2 **£59**

Library Source Kits available in all languages

Call JPI on (0234) 267500 now for your free copy of the TopSpeed 1992 Compiler Catalogue

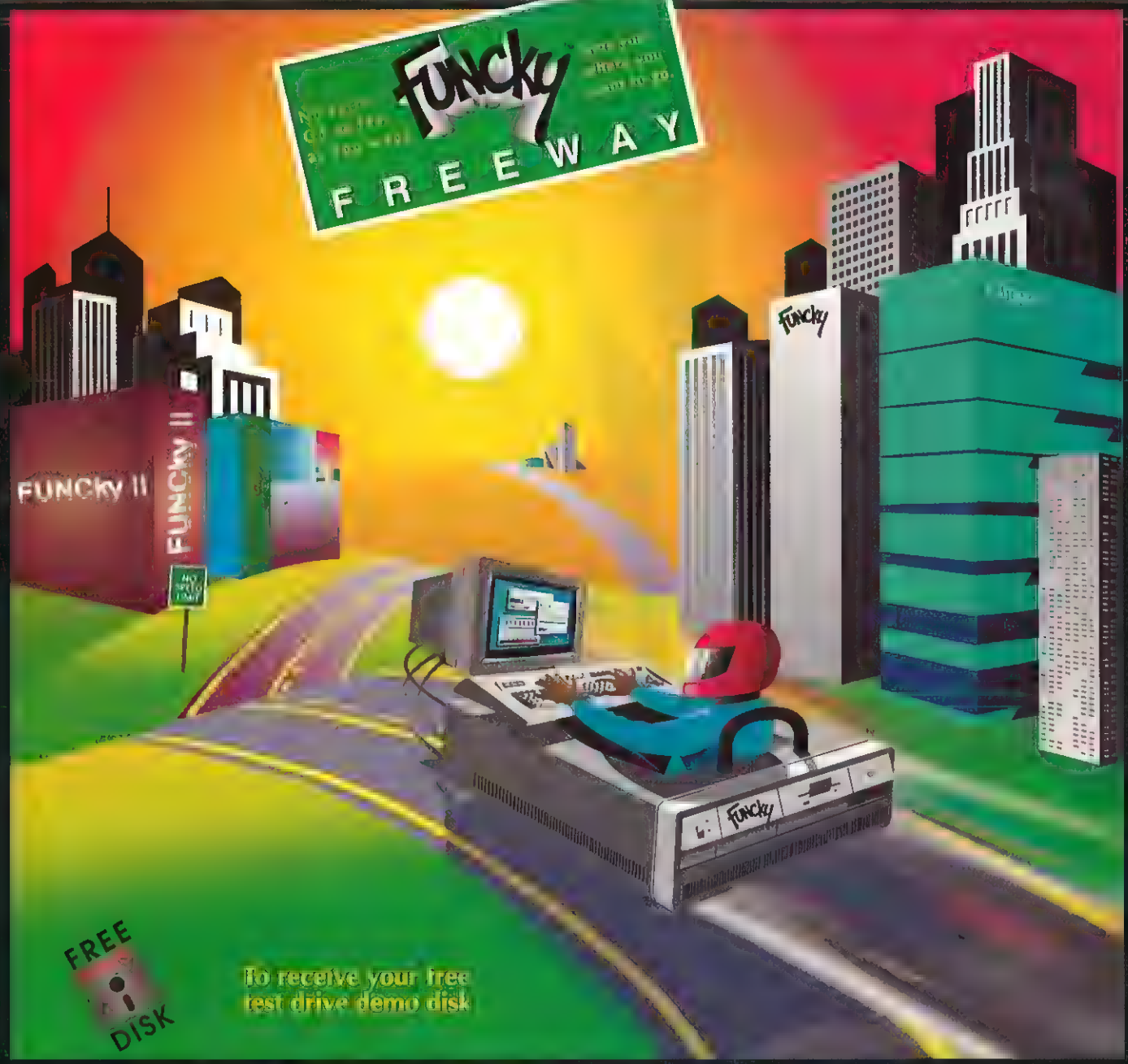
TopSpeed products are available from:
GreyMatter 0364-53499 · System Science 071-833-1022
RTA 081-656-7333

JPI
3 The Mansards, Tavistock Street, Bedford MK40 2RX
Fax: (0234) 217094

TopSpeed® C++ TopSpeed® Modula-2 TopSpeed® Pascal TopSpeed® C

CIRCLE NO. 914

FREE TEST DRIVE!



Try our innovative TEST DRIVE program, a free demo of the FUNCky library which you can actually program with. Edit code using any or all of the FUNCky library, then compile, link and run to find out for yourself why FUNCky is the ultimate Clipper Developer's tool.

The FUNCky add-in library for Clipper contains over 450 functions which can be used as if they were a part of the Clipper language itself. Extensive features include moveable resizeable windows with mouse support, superior DOS file handling, blazing fast string manipulation, and impressive video control.

Get FUNCky now and get the phenomenal programming power you need to create the advanced applications demanded of you in today's competitive development environment.

dLESKO
INCORPORATED
320 York St., Jersey City, NJ 07302
FAX# 201-435-0637

Contact QBS Software

Telephone: +44 81 994 4842
Fax: +44 81 994 3441
BBS: +44 81 747 1979

CIRCLE NO. 915

requirement for an operating system to convert the PC hardware from personal computer to workstation is a multi-tasking, multi-session, windowed system which will execute existing DOS programs alongside new applications. This constraint is a strategic mistake for which we are paying dearly, although it is an understandable mistake.

The first problem encountered was that the memory manager of the 286 is inadequate to run virtual 8086 sessions and so a 386 SX is the minimum hardware requirement.

The second problem with running DOS programs in a windowed environment is that they are still all different. It is only properly written programs that will integrate and hence reduce training and support costs. In the end it is cheaper to throw the DOS programs away and to replace them with consistent, integrated equivalents.

Client/Server Systems

While UNIX derivatives have been successful for engineering workstations, they consume considerable resources. There is also a fear of UNIX and its user hostile interface (no worse than PC DOS, but believed to be so) so that it has been rejected for desktop PC applications. Users wanting UNIX on a workstation are better served by low end RISC machines. UNIX on a PC is dominant for multi-user, VDU-based systems, particularly small business systems.

Since IBM and Microsoft controlled the PC market with MS-DOS, and this suited software vendors such as Novell, Lotus, Borland, and so on, progress was stopped for most of the 1980s. Mass advertising brainwashed PC users into accepting inferior products (excellence in mediocrity) despite the lessons taught by the Mac, Sun, etc. But the need to integrate workstations with data bases reached most corporate users attention by the end of the 80s, heralding the current interest in client/server computing, with both PC, UNIX and mainframe database servers.

File serving is adequate for 'multiple single-user' personal computing, but corporate requirements involve multi-user, groupware etc and must provide shared database services. SQL requestors are available for MS-DOS and DOS + Windows 3, but this unprotected conglomeration of software entities posing as an operating system is technically unsound. It can never be made safe against erroneous application software and thus can never be made acceptable.

Despite claims to ongoing development of DOS/Windows, this is mythology. It must be replaced by a properly designed OS, which is now possible due to the working memory manager of the 386/486 family of processors. The memory manager is the most important feature, but the power and the 32-bit linear addressing will make native applications far superior and far cheaper than DOS equivalents.

OS/2

The first attempt to provide the industry standard PC operating system, OS/2 version 1, was a complete failure. IBM and Microsoft may be the best marketing companies in the world, but their technical competence leaves much to be desired. The product was based on the wrong processor, the 286, and failed to provide any usable desktop front-end or any programming tools worth having. It wouldn't run DOS programs, and there were few OS/2 1.x programs and no tools to write them with. IBM provided the GUI with Presentation Manager to embed in the basic OS, but then attempted to ship a proprietary version, OS/2 extended edition, by adding the communications and database managers.

The folly of the 286 processor design was quickly realised and work began on the 386 (32-bit) version 2. Beta versions were tested as far ago as April 90 (Byte Magazine). With no real knowledge of what actually happened it appears that Microsoft's OS/2 2.0, even though they had shipped a beta version, didn't work and IBM took over development.

Microsoft reacted to the loss of OS/2 (and its future!) by releasing its 'poor man's Mac' equivalent, Windows 3. By wooing clone makers, who hoped to sell the 4 to 6 MB memory upgrades Windows 3 needs, it has been an enormous success for Microsoft. At the same time it is a potential disaster for the corporate users, since its technical shortcomings are not solvable and thus it will result in even more expensive PC systems. Windows 3 does provide superior facilities to DOS and thus specific Windows 3 applications are desirable. As a result the requirements for the real PC operating system must now include the ability to run 16-bit Windows 3 applications as well.

Thus, very late in the day, April 92 represents a milestone, if IBM has got the product technically right, ie it is robust, performs well and is economic. OS/2 2.0 provides the system users want; ask a committed Windows 3 user to describe what he or she wants from the system and they will accurately define OS/2 2.0, not Windows 3.

Windows NT

Despite its current euphoria over Windows 3, Microsoft is not stupid. It has brilliantly sold an inadequate software package, which must be replaced. Those users who say they won't upgrade should think of how users weren't going to upgrade from XTs to ATs or from 286 to 386 machines - but they did. Thus there are 10 million and more replacement operating systems to be sold, which will likely be OS/2 2.0. Microsoft has predictably reacted by talking

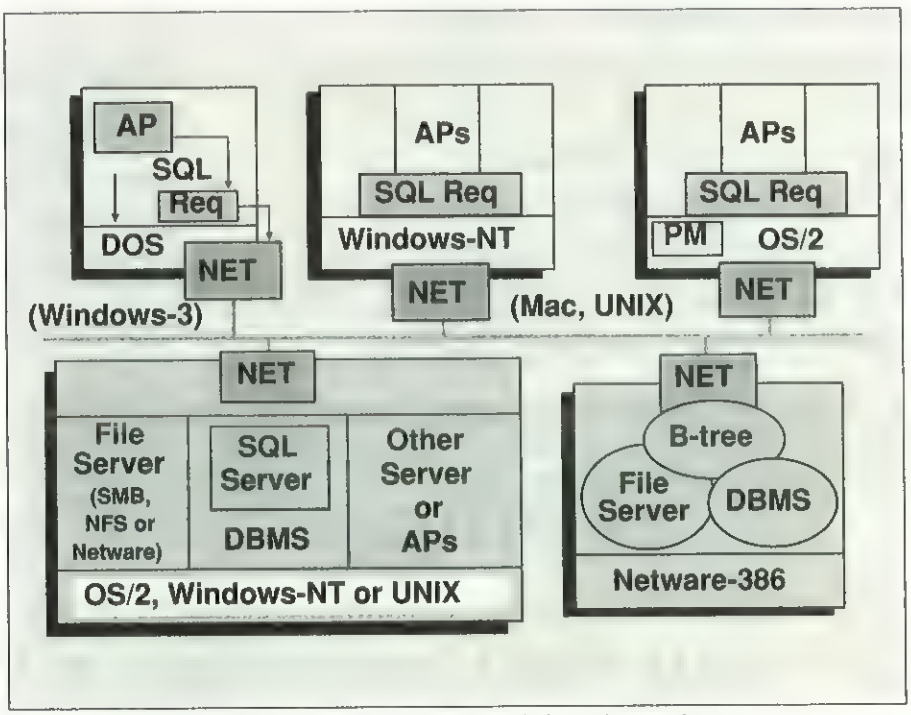


Figure 2 - Third Generation (Client/Server) LANs

COMPUTER LANGUAGE PRODUCT EXCELLENCE AWARD 1991 COLA

PC-lint 5.0 presents C Bug # 564

```
int C_greater_than_Cplusplus( int C )
   {
   return  C > C++ ;
   }
```

Although this looks like a comparison of two well known programming languages, it is really a C function with a subtle error. Can you or your compiler spot it? Call if you need a hint. Refer to Bug #564.

PC-lint will catch this and many other C bugs. Unlike your compiler, PC-lint **looks across all modules** of your application for bugs and inconsistencies.

New – Optional Strong Type Checking and variables possibly not initialized.

More than **330 error messages.** More than **105 options** for complete customization. Suppress error messages, locally or globally, by symbol name, by message number, by filename, etc. Check for **portability** problems. Alter size of scalars. Adjust format of error messages. Automatically generate ANSI prototypes for your K&R functions.

Attn: Power users with huge programs.

PC-lint 386 uses DOS Extender Technology to access the full storage and flat model speed of your 386. Now fully compatible with Windows 3.0 and DOS 5.0

PC-lint 386 – $239
PC-lint DOS - OS/2 – $139

Mainframe & Mini Programmers

FlexeLint ***in obfuscated source form, is available for Unix, OS-9, VAX/VMS, QNX, IBM VM/MVS, etc. Requires only K&R C to compile but supports ANSI. Call for pricing.***

Gimpel Software

3207 Hogarth Lane, Collegeville, PA 19426

CALL TODAY (215) 584-4261 Or FAX (215) 584-4266

30 Day Money-back Guarantee.

PA add 6% sales tax.

PC-lint and FlexeLint are trademarks of Gimpel Software

CIRCLE NO. 916

about Windows NT, which it hopes will slow down the take-up of OS/2 2.0 and then win the bulk desktop workstation market.

Windows NT is well specified. It is based on microkernel technology, akin to Mach (as adopted by OSF) and Chorus (UNIX V.4), which means that it can support multiprocessors, guest operating systems, specialised servers etc. Two initial versions are proposed, one for the Intel 386/486 and the other for the MIPS RISC processor, part of the dubious ACE consortium. A specific version of Windows NT, possibly called Win32, will vie with OS/2 2.0 for the PC workstation, but other versions will vie with UNIX for server and engineering applications.

With Windows NT Microsoft are really taking the whole world on, head on.

Microsoft Versus IBM

A key aspect of the OS/2 2.0 market is the alliances that keep cropping up in our disjointed industry. Now that IBM have fallen out with Microsoft, which can only be to the good of the industry, Microsoft is looking to other partners who are antagonistic towards IBM. DEC is the current target to replace IBM in the Microsoft camp, a significant player, but an ailing one and still nothing like as big as IBM. DEC is making noises about an interest in Windows NT, but it is also rumoured to be lining up with IBM for OS/2 2.0 (backing both horses, in line with the Japanese and most others). In fact, DEC is particularly interested in Microsoft Lan Manager, the basis for Digital Pathworks, rather than any specific OS.

In the 1980s Microsoft could, and did, produce any product, which backed up by IBM couldn't fail to succeed. The unholy alliance was a virtual monopoly. Now Microsoft must not only make it alone but in competition with IBM.

Conclusion

Unless IBM has made a mess of OS/2 2.0 technically, it will surely succeed in its own backyard: the corporate IBM die-hard mainframe sites. They need OS/2 2.0 to gain client/server computing in an SNA world with a mainframe DB2 database.

But IBM must be favouring winning the bulk of Microsoft's considerable revenues by dominating the DOS and Windows 3 replacement market. To do this it must woo all PC clone suppliers, including rivals such as DEC and H-P, in direct competition with Microsoft. This implies a completely new PC sector stance from IBM. Gone is the concept of a 'vertical' proprietary PS/2 loaded with OS/2 EE workstation. IBM must commit that there will be no specialised IBM versions of OS/2 2.0 and the PS/2 must compete as it does today as another PC.

IBM has also made strategic alliances with key PC industry software suppliers Novell, Lotus and Borland. They will be shipping true 32-bit OS/2 2.0 versions of their products which will inevitably be superior to the 16-bit Windows 3 versions. Only Microsoft will be favouring Windows NT over OS/2 2.0, although everyone will once again back both horses, just in case OS/2 2.0 fails.

One key to the success of OS/2 2.0 must lie with the wealth of applications available for it. In sharp contrast to OS/2 1.x, which had no applications, OS/2 2.0 will have all the DOS and Windows 3 programs plus the new 32-bit versions. There are also numerous tools, eg Choreographer, Smalltalk V, for developing in-house applications.

The ideal solution for everyone will be a choice of OS/2 2.0 or Windows NT, with relief from Windows 3, by the end of 1992. A choice of two operating systems leads to competition and ongoing improvements, as demonstrated in the UNIX world by the success of OSF/1 versus UNIX V.4. Unfortunately serious doubt must be placed over the possibility. OS/2 2.0, yes, but can Microsoft get Windows NT beyond the beta stage? Its track record says it will be two or three years yet before it has an acceptable release, by when it will be too late. It would be foolish and naïve to believe any of Microsoft's claims; it is beholden to prove that it can, for once, meet its claims.

By the end of '92, my preferred LAN system will be DOS (286) and OS/2 2.0 (386/486) workstations with UNIX servers on RISC hardware, using TCP/IP protocols. I will favour a strategy based on the replacement of existing LANs, rather than one based on continuity.

Finally: Dear IBM, please could we have a new name. OS/2 2.0 is a mouthful and could be confused with that awful Microsoft product OS/2 1.x, which is better quickly forgotten!

EXE

Martin Healey is Chairman of Technology Concepts Ltd, Emeritus Professor of The University of Wales and Vice President of the Institute of Data Processing Management.

FORGET THE REST GET THE BEST in Clipper® productivity

GRUMPFISH LIBRARY . . .

Grumpfish Library makes your applications more powerful and more appealing, while cutting down on your development time. You will not have to waste hours learning a new language to use Grumpfish Library — even new Clipper developers can integrate Grumpfish modules into their programs within minutes of breaking the shrink-wrap.

Grumpfish Library is written 99.9% in Clipper, and all 5.01 and Summer '87 source code is included (nearly 900K worth). Not only can you modify the source code to suit your specific needs, but it will inspire you to produce more efficient and powerful code of your own!

Grumpfish Library is not merely 5.0 compatible — it has been **completely rewritten from the ground up** to take advantage of all the powerful Clipper 5.01 features.

Grumpfish Library Features:

- Pop-up desktop utilities - spreadsheet (NEW), calculator, calendar, appointment tracker, notepad, stopwatch
- Numerous extensions to the GET command (CALCULATOR, LIST, PROPER, etc.)
- Generic database browser with automated record layout add edit view QBE routines. Includes a built-in screen painter that lets you design and generate Clipper code for your data entry screens in seconds!
- Nearly 100 functions to make your Clipper development life easier
- Dynamically overlayable by the current crop of dynamic overlay linkers
- 30 days free voice support, one year unlimited free BBS support
- Handsome printed documentation, Norton Guides reference database
- No hassle unlimited royalty-free integration license

GRUMPFISH MENU . . .

Slash your development time with Grumpfish Menu! Create working prototypes in minutes instead of days. Handle change orders in seconds instead of weeks! If you can create a text outline file, you can create a gorgeous, easy-to-use front end with Grumpfish Menu — it's that simple!

There is no need to learn a template language . . . just store your menu structure in a text outline file, and Grumpfish Menu instantly generates optimized, ready-to-compile Clipper source code for either Summer '87 or 5.01! Now you can change your menu structure as much as necessary without having to worry about screen housekeeping or case logic.

Grumpfish Menu is shipped with two versions of the menu generator and linkable libraries: one for Summer '87 and one for Clipper 5.01. The supporting functions for the menu system are written 96% in Clipper. (Assembler is used only for transparent shadowing and certain string handling/DOS functions.)

Grumpfish Menu Features:

- Seven menu styles, including pull-down, cascading, 1-2-3, and boxed 1-2-3
- Hot keys on any menu item (NEW)
- Security levels on any menu item (NEW)
- Mobile and resizable menus (NEW)
- Built-in tutorial system
- Unprecendented aesthetic control — change menu attributes **from within your application without recompiling or relinking!!**
- User-defined configuration files can be loaded at run-time on the command-line — each user can have their own interface!
- 30 different configuration options
- Develop quick prototypes by toggling one configuration option. Grumpfish Menu will ensure that calls to non-existent functions will be handled gracefully, rather than crashing your program.
- Embed source code anywhere within your menu structure.
- Define your own custom entrance and exit routines.
- Allow users to toggle menu options or tag multiple options by adding one line to your menu outline file.
- Seamless support for Grumpfish Library desktop utilities and Dr. Switch-ASE™
- 100% compatible with the current crop of dynamic overlay linkers
- 30 days free voice support, one year unlimited free BBS support
- Handsome printed documentation

Available through

QBS SOFTWARE

10 Barley Mow Passage, Chiswick,
London W4 4PH Tel: 081 994 4842
Fax: 081 994 3441 BBS: 081 747 1979

081-994 4842 Call now for a free demo disk

GRUMPFISH, INC.

2450 Lancaster Dr. N.E. Suite 206 Salem, OR 97305
Tel (503) 588-1815, Fax (503) 588-1980 BBS (503) 588-7572, CompuServe 70673,355

Grumpfish — Your Guide to Clipper Education and Productivity

CIRCLE NO. 918

CodeBase++

How do you provide a consistent C++ interface to your C library? Paul Kemp finds out what Sequiter Software did with CodeBase.

Way back in March '91 Bryan Boreham wrote an article in *.EXE* entitled *Adding a C++ front-end.* As a C++ programmer using a C-based library, he described the design and implementation of a number of C++ 'wrapper' classes to furnish the CodeBase 4.0 library with an object-oriented interface.

CodeBase itself has come a long way since then, and in version 4.5 Sequiter Software has included its own C++ interface. As library vendors are under increasing pressure to provide support for the rapidly expanding C++ market, how has Sequiter fared in its attempt to design 'A Library For All Seasons'?

C++ class name	*C structure name*	*Description*
c4Code	C4CODE	Global settings
d4Data	D4DATA	Data file information
d4Date	None (char[8])	Date information
f4Field / f4Memo	F4FIELD	Fields
i4Index	I4INDEX	Index file information

Figure 2 - C++ classes and C structures

Overview

CodeBase is a comprehensive library of some 185 C functions that allows programmers to create, interrogate and update dBASE files. Version 4.5 of the library supports the dBASE III, IV, FoxPro 2 and Clipper 5 data, index and memo file formats. For dBASE IV compatibility you can choose to use multiple tag (.MDX) indexes, and for FoxPro 2 the new compound (.CDX) index files.

The library is written in ANSI C and full source code is provided, so that it is possible to port the code between DOS, Windows, OS/2 and UNIX simply by recompilation. Semi-automatic record and file locking capability is also built into the library so that it can be used in any number of network environments (eg Novell, 3COM, LAN Manager and UNIX).

Most standard C and C++ compilers are supported including Microsoft, Borland, Zortech and Watcom. There are pre-built libraries for the most popular compilers but it is relatively trivial to build your own custom libraries with batch files that are provided (I was using Borland C++ 3.0 and had to rebuild the supplied Borland libraries because they were created using version 2.0). Support for Visual Basic and Turbo Pascal for Windows is achieved by the provision of a Windows DLL.

dBASE itself is not required so that standalone applications can be built, and there are no royalties; executables and the CodeBase 4.5 DLL can be distributed free of charge.

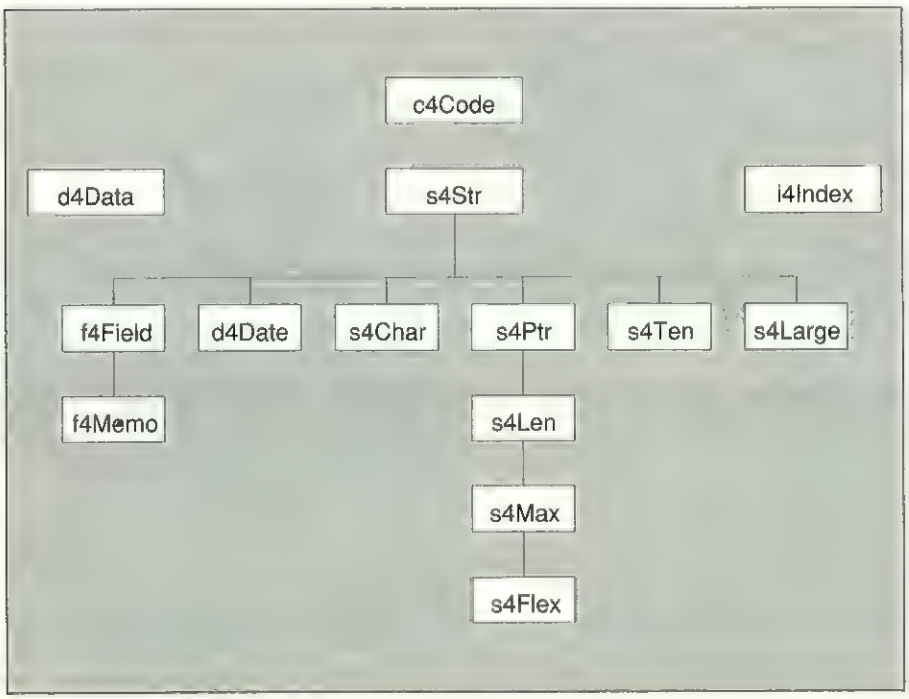


Figure 1 - CodeBase C++ class hierarchy

The C++ interface

Figure 1 shows the CodeBase C++ hierarchy. In an otherwise excellent set of documentation, I was somewhat miffed to discover that such a simple representation of the hierarchy was nowhere to be seen. I have noticed the absence of such diagrams in the documentation of some other C++ libraries and have to say that I find it quite unforgivable; it is the starting point for a C++ programmer getting to grips with a new library.

The CodeBase C library is designed in an 'object-oriented C' fashion that has become

```
#include <d4all.h>
#include <d4data.hpp>

F4FIELD_INFO cust_fields[] =
{
  { "NUMBER", 'N', 8, 0 },
  { "NAME",   'C', 20, 0 },
  { 0, 0, 0, 0 }
};

char * namelist[] =
{
  "Major",
  "Kinnock",
  "Ashdown",
  "Sutch"
};

void main()
{
  c4Code c;
  d4data d;

  // create CUSTOMER.DBF
  d.create( &c, "CUSTOMER", cust_fields );

  d.append_start(0);
  f4Field number( d, "NUMBER" );
  f4Field name( d, "NAME" );

  // append records
  for (long i=0; i<4; i++ )
  {
    number.assign_long( i, -1 );
    name.assign( namelist[i] );
    d.append();
  }

  // print records
  for ( d.top(); !d.eof(); d.skip(1) )
  {
    printf("\n");
    for ( int j=1; j<=d.num_fields(); j++ )
    {
      f4Memo field( d, j );
      printf( " %s", field.str() );
    }
  }
}
```

Figure 3 - Create any print file

more popular in recent years. Most of the functions are grouped in such a way that the first parameter is a pointer to a `struct` which contains all the basic information that the function needs. This type of design facilitates the provision of a C++ interface as the data structure corresponds to a class' `private` data items, and C++ takes care of the pointers with `this`. Figure 2 shows the relationship between CodeBase's C++ classes and its C data structures. For example, the C code fragment

```
C4CODE c;
e4exit_test( &c );
```

in C++ becomes

```
c4Code c;
c.exit_test();
```

Except for class `f4Field`, member functions of the C++ classes are inline calls to the corresponding C function, so .EXE sizes and performance should be approximately the same as using C.

However, there is more to C++ than data abstraction, and it is inevitable that any C++ hierarchy that is based on an existing C library will suffer because it is limited in the number of object-oriented features that can be employed. This is certainly true of the CodeBase C++ interface but, having said that, the company has done a pretty good job.

Fields and records

Where a C++ programmer gains the biggest advantage is in the use of the `s4Str` component hierarchy. Because the `f4Field`, `f4Memo` and `f4Date` classes are all derived from `s4Str` they can be treated as strings. So instead of having to use myriad C functions to manipulate field data, operator overloading and function overloading have been used to good effect in simplifying application code enormously. Figure 3 shows a simple application which creates, populates and lists the contents of a .DBF file called 'CUSTOMER'.

For each data file opened, CodeBase allocates memory for one record worth of information. Each of these memory areas is called a 'record buffer'. Many of the high-level functions, such as `d4Data::go()` and `d4Data::skip()` automatically use this memory buffer. It is the field functions which manipulate information in the record buffer. If data in the record buffer is changed, CodeBase flags the record as having been modified and automatically writes the changed information back to the data file when the contents of the record buffer are flushed (eg by reading the next record with `d4Data::skip(1)`). The following code fragment assigns 'Bobby' to the NAME field of the first database record in file 'CUSTOMER':

```
c4Code c;
d4Data d( &c, "CUSTOMER" );

d.top();
f4Field name( d, "NAME" );
name.assign("Bobby");
d.skip(1);
```

CodeWindows

Bundled with CodeBase 4.5 comes version 1.0 of CodeWindows. It comprises a set of Windows-specific C functions that greatly ease the writing of data entry, edit and browse windows. Dialog boxes are designed in the normal way and control IDs are then linked to database fields in the application. The CodeWindows functions then do most of the hard work for you so that simple applications can be put together in minutes. Picture templates and user-defined validation functions can be defined for edit controls and a basic print module is also supplied.

I was very impressed with this part of the library although, unlike the standard database functions, there is not yet a C++ interface. When this is provided it will prove to be an extremely powerful tool.

Conclusion

CodeBase 4.5 is one of the most flexible libraries that I have come across. Through good design, Sequiter has managed to put together a set of functions that can be used on a wide variety of platforms and in many programming environments. To be sure, compromises have been made, but these have been kept to a minimum through the application of object-oriented techniques. CodeBase is not an object-oriented database - it is operating on dBASE files, but the C++ wrapper works well and sets a good example for other C library vendors.

EXE

CodeBase 4.5 with CodeWindows 1.0 from Sequiter Software is distributed in the UK by The Software Construction Company (tel 0763 244114) priced £255 + VAT.

Will your application fit the right environment?

Developing a graphical user interface can take up to ninety per cent of the effort it takes to build a complete application. Duplicating that work each time an application is moved to a new environment is simply not cost effective.

In today's multi-platform world portability is essential. Aspect is the answer.

Aspect is the only user interface tool you'll ever need and it's available now exclusively from Gold Mine Technologies.

With Aspect, your application can run on any windowing system and platform, including Unix with Motif and Open Look, Microsoft Windows, Macintosh, OS/2 Presentation Manager and character terminals.

Aspect is a powerful, easy to use development system that stores user interface information independently of program logic. Moving between environments is simplicity itself.

With Aspect's potent development environment, building windowing applications is fast and straight forward. Aspect is easy to learn, and once learnt, is the only windowing development system you need.

For more information about Aspect contact Malcolm Neill or John Rodford on 0442 862824 or fax: 0442 876296 or fill in the coupon below and send to:

Gold Mine Technologies Ltd.,
1 Church Gates, The Wilderness, Berkhamsted, Herts HP4 2AZ.

aspect *The* future *of* user interface technology

CIRCLE NO. 919

☐ Please send me more information about Aspect
☐ Please call me to arrange a meeting

Name

Position

Company

Address

Postcode

Telephone

La résistance

In 1992, the power of C++ dominated the known programming world. All the world? Not quite. For in a small Gallic village... Willie Watts talks to chief Eiffel Druid Bertrand Meyer.

Can you give us a sketch history of yourself and Interactive Software Engineering?

I am originally a pure product of the best that the French system has to offer in terms of general, scientific and engineering education. This was completed by an MS in computer science at Stanford in the US.

At Stanford I didn't have enough time to learn much in detail, but I learned what was important and what was not. So that after that I was able to read a lot and learn by myself. I was already working in industry, but at least what I learnt what was important from Knuth, McCarthy and a few people like that. I also learnt about object-oriented technology at Stanford. I was fortunate enough to run into a description of Simula and it struck me right away as the way to program.

After that I came back to France, and went to work for a company called 'EDF', which is the equivalent of what the Central Electricity Generating Board used to be in Britain. I was head of a group which was in charge of software engineering, programming education, software engineering tools, libraries and also links with research in software engineering.

I had a kind of dual role - partly operational (very operational, actually), but also at the same time research. The non-academic side involved, among other things, setting up a training programme on modern software methodology, which opened my eyes to the reality of industrial software development. At the same time I was quite active in research in programming methodology. I wrote a book called *Méthodes de Programmation*, ie 'Programming Methods', which was very important for me and was also quite influential in France. It was used in the main educational institutions, and it put me in contact with professors.

Bertrand Meyer and Eiffel

With the current plethora of object-oriented languages and tools, and C++ apparently set to conquer the Universe, the reader might reasonably ask why *.EXE* chooses to devote so much space to the inventor of yet another language - a language which is still comparatively unknown in the wider programming community.

The answer, in a nutshell, is Meyer's book *Object-Oriented Software Construction* (pub. Prentice-Hall, ISBN 0-13-629049-3). This book, I feel I can state with little danger of effective contradiction, contains the most lucid description of object-oriented techniques that you will find; whatever their opinion of the Eiffel language, few of Meyer's opponents will deny that he can write like an angel.

The book has acted as a superb advertisement for Eiffel. Interest in the language has spread from the academic world, and is now beginning to appear in the commercial mainstream. At the same time, control of Eiffel has been passed from Meyer's company Interactive Software Engineering to a non-profit consortium, and alternative (non-ISE) implementations of the language have begun to appear, notably SIG's implementation for DOS. Eiffel is fast becoming a practical possibility for ordinary projects.

Eiffel is different. It is not just another Algol variant with a few bolted-on object extensions. Programmers who track current thinking should at least to know about the ideas in Eiffel; this interview may serve as a taster for further reading. ***WRW.***

Santa Barbara

I had this dual career for about eight or nine years. In 1983, I decided to develop the academic side of me a little further, so I took what I thought would be a year's sabbatical at the University of California Santa Barbara (in the end I stayed four years). There I taught much of the basic software curriculum.

Now, in the data structures and algorithms course that I taught it was the department policy that the professor should use C. This was strange, because most of the other courses enjoyed a lot of academic freedom. But C was a particular requirement because they were using this course for two purposes: 1) the official one - to teach data structures and algorithms - and 2) to serve as a filter to separate the men from the boys.

You're saying the course was made deliberately difficult?

Oh yes. Deliberately puzzling. I hated it. I should say that I didn't know C very well at the time, so I had to learn it - and it was horrible. I wanted to teach data structures, algorithms, and systematic approaches to programming problems. Instead of that, I found myself trying to help students debug programs using pointers, finding incorrect memory references and so on. I decided that I could never do that again.

So in '85 I started Interactive Software Engineering with a few colleagues in order to follow up certain aspects of our research. The work in the company soon became more interesting, from a purely scientific viewpoint, than the work we were doing in the University, because I could use whatever I found scientifically valuable as opposed to bowing to the student pressures to use C.

When did Eiffel appear?

When we started the company we really wanted to develop some CASE tools - in particular a structural editor, now the Archi-Text product. But I also wanted to use proper techniques - I didn't want to be too far from what I was trying to teach. So we looked around for a decent object-oriented environment and we just couldn't find one. Smalltalk was attractive, but it was too far from the mainstream. C++ was available (as was Objective C) but we really didn't discuss it. It was not up to our standards.

I very quickly wrote a specification for a modern version of Simula, with some simplifications and some extensions. I also a wrote a basic version of the libraries. Libraries are the key to the success of object-oriented technology in Eiffel. If it can be defined by just one sentence, Eiffel is a language to write the Eiffel libraries - that is to say to write the best possible, reusable industrial-quality software components that we could think of.

I did that in just a week or so, and that was it. At first, we didn't really think about it as being a tool for others. It was only months later that we realised that we had something that no-one else really had: a complete, statically-typed, efficiently implemented language that could actually provide an object-oriented solution to mainstream industry. It's only then that we started thinking about marketing the product.

How was Eiffel first implemented? Was it one of those languages which was always implemented in itself?

No. We didn't have a bootstrap strategy initially, which may have been a mistake. We just used the obvious solution, that is to say to write it in C. And I should say we were sorry for that decision for a long time. It is only in Version 3 that we have managed to write the entire system in Eiffel. With our portability requirements, C was the almost inevitable choice.

What's in a name?

Why is it called 'Eiffel'?

The question should be, 'How could it be called anything else?' It was almost inevitable. First there's this habit of naming languages after people: Pascal, Turing and many others. We decided, for once, to take not a scientist but an engineer, and a really great engineer. Second reason: the method promotes bottom-up software construction. If you think of the shape of the Eiffel Tower, and you try to build it top-down... Third, and probably the important: if you look at that structure, it's an extremely elegant, powerful, reusable, extensible structure. However it is built of a few small, simple parts which you combine, which is exactly the same idea as object-oriented software construction.

The opposition

How would you characterise the difference between Eiffel and C++?

There is a difference in philosophy and there are technical differences. The difference in philosophy comes from a different view of what 'compatibility' means. I don't think there is any disagreement with respect to the necessity of reusing existing software, particularly C software. The divergence resides in how it should be done.

The C++ (or Objective C, or Turbo Pascal) approach is that you should have compatibility at the language level. So you take C and you add things to it. The Eiffel view is that compatibility with existing software is not an excuse for polluting the language. The language can be C, which is a certain technology, or it can be object-oriented, which is completely different.

You don't make a device that is both a diode and a transistor. If you have an existing machine that uses diodes then, fine - you want to keep it. You could put in wires to the transistors and have communication links and so on, but you don't try to transform a diode into a transistor incrementally. This is the Eiffel position. You choose between the Eiffel, object-oriented world and the C world, but you should keep the two separate; because otherwise you risk losing on both counts.

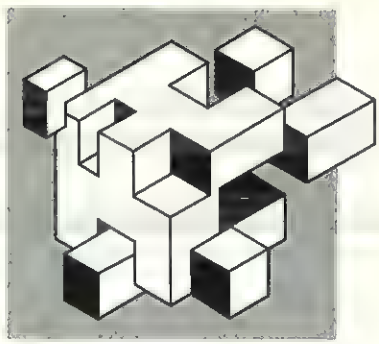
You lose the simplicity of C: you lose the ability to implement it efficiently, the ease of writing it and the ease of teaching the language, which are the three major advantages of C. On the object-oriented side, you again lose a lot. You lose simplicity - because the object-oriented paradigm has to be combined with totally incompatible concepts. You lose the ease of teaching the object-oriented paradigm. And you lose some of the most important advantages of object-orientedness. For example, it is quite impossible to have garbage collection in C. You lose typing - you cannot have both the C type system and an object-oriented type system. You lose the ability to do things like exception handling properly, as you must take into account all kinds of bizarre side-effects.

So you actually lose the most important benefits, the real breakthroughs, the quantum leaps that you can get from object-oriented technology, and all because of this stupid requirement of remaining compatible with something that has nothing to do with object-oriented technology. The technology is too good, too important, too potentially beneficial to damage it because of concerns that may appear valid in 1986, or 1989, or 1992, but will totally disappear from the scene if the technology becomes successful.

As for technical differences, there's a whole list. Type-checking; assertions; genericity (particularly constrained genericity, which is the only way to get safe genericity); exception handling (which has been proposed for C++, but Eiffel has them now),

SOFTWARE DEVELOPMENT 92

The Software Tools Show



16-18 JUNE 1992, WEMBLEY CONFERENCE AND EXHIBITION CENTRE, LONDON.

TECHNICAL CONFERENCE PROGRAMME

Calling Software Developers...

Do you need to keep abreast of the latest technological advances, so as to increase both the functionality and efficiency of your applications development work?

Are you responsible for strategic, mission critical systems development?

If the answer is 'yes', you cannot afford to miss the Software Development 92 Technical Conference. Benefit from the expertise and guidance of leading industry figures who will advise you on the latest technical advances in systems/software development including: **Windows, UNIX, OS/2, CASE** and **OOPS** development.

This is your invitation to be a part of the UK's only dedicated software development event - helping to shape and give voice to one of the major growth markets in computing today.

TUESDAY 16TH JUNE

OPEN PLENARY – setting the scene

THE INSTITUTE OF SOFTWARE ENGINEERING

09.30 The ever increasing importance and scope of CASE. Paul Madden, Research & Consultancy Manager, The Institute of Software Engineering.

10.00 CASE integration - more than having a repository. Ken Thompson, Research and Consultancy Manager, The Institute of Software Engineering.

10.30 Reverse engineering - is it worth it? Alan Frazer, Consultant, The Institute of Software Engineering.

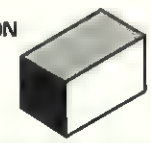
11.00 Coffee

11.30 Object orientation – not just for the programmers. Dr George Wilkie, Consultant, The Institute of Software Engineering.

12.00 Application generators - fact or fiction? Dr Patrick McParland, Consultant, The Institute of Software Engineering.

12.30 Group discussion

13.00 LUNCH AND EXHIBITION VISIT



TRACK 1

DATABASE ARCHITECTURES & PROGRAMMING.
Chairman – Richard Attenborough, Editor, ProgramNOW.

14.00 Testing 4 GLs - a comparison. Christine Seager, Systems Analyst, Loud and Bow.

14.45 User requirements have led to OODBM's. Mark Ellis, Technical Support Manager, Sema Software Technology.

15.30 Tea

16.00 Things in my notebook .... an object oriented database for small computers. Steve Swallow, Quality Assurance, System Applied Technology.

16.45 Multimedia design for databases. Sebastian Karl, Sales Manager, Continuous Technology.

17.30 Close

TRACK 2

UNIX
Chairman - Mr Zdravko Podolski, Project Manager, UK Unix Users' Group and Insignia Solutions.

14.00 The future of Unix. Chris Stott, Kernel Engineering Manager, The Santa Cruz Operation.

14.45 Unix as the future. Graham Taylor, Chairman UK Marketing Group, Unix International.

15.30 Tea

16.00 What is SVR4? Nick Price, Technical Director Europe, Unix International.

16.45 NeXtStep and UNIX. Gregor Bailar, Managing Director, NeXt Technology.

17.30 Close

WEDNESDAY 17TH JUNE

TRACK 1

NETWORKS ARCHITECTURE & OPERATING SYSTEMS
Chairman - Richard Attenborough, Editor, ProgramNOW.

08.30 Distributed Computing: Models technology for real-world solutions. Richard Morley, Consultant Analyst, Systems Support Associates.

09.15 Enterprise-wide client/server computing. Neil Morgan, Desktop Product Marketing Manager, Oracle.

10.00 Coffee

10.30 XBase - the issues. Speaker to be confirmed.

11.15 SQL on power - PC networks. Adrian Lea, Freelance Analyst.

12.00 LUNCH AND EXHIBITION VISIT

EXPERT SYSTEMS
Chairman - Nick Evans, British Computer Society Specialist Group on Expert Systems.

14.00 GUI building & AI - an alternative approach. Richard Barker, Development Manager, Scientia.

14.45 Generic algorithms in CASE. Alistair Blair, Development Manager, ITACS.

15.30 Tea

16.00 Intelligent Systems. Matthew Colbourne, Business Manager, Advanced Technologies Group, SD-Scicon Consultancy.

16.45 Distributed agents in parallel data fusion. William Pickles and David Catton, Directors, Strand Software Technologies.

17.30 Close

TRACK 2

OOPS DESIGN
Chairman - Chris Stone, President, Object Management Group, USA.

08.30 Object oriented method - adoption and selection. Alec Bruty, Director, Axon Seminars and Training.

09.15 Adopting an object oriented approach. Nicholas Wybolt, Director of Engineering, Cadre Technologies.

10.00 Coffee

10.30 The object design challenge. John Daniels, Director, Object Designers.

11.15 Object oriented databases. Mary Loomis, Vice President, Versant Object Technology, USA.

12.00 LUNCH AND EXHIBITION VISIT

OOPS PROGRAMMING
Chairman - Chris Stone, President, Object Management Group.

14.00 Options in object oriented development. Steve Baker, Object Business Manager, QA Training.

14.45 OOPS design metrication. Dr Trevor Hopkins, Senior Lecturer, Department of Science, Manchester University.

15.30 Tea

16.00 The silver anniversary of the silver bullet. John Salt, Simulation Software Engineer, Eurotunnel.

SPONSORED BY

1980 BCS Specialist Group in Expert Systems 1990

ECUG

ProgramNOW

SOFTWARE DEVELOPMENT MONITOR

THE INSTRUCTION SET
Hoskyns Open Systems Training

16.45 Investing for the future with object oriented Clipper. Alan Davies,Technical Manager, and Roy Corneloues, Technical Marketing, Nantucket.

17.30 Close

THURSDAY 18TH JUNE

TRACK 1

THE HUMAN COMPUTER INTERFACE

Chairman - Miles Macleod, Senior Scientific Officer, HCI Group, National Physical Laboratory.

08.30 Migration to graphical user interfaces in the 90s. Mark Rogers, Managing Director, MF Systems.

09.15 HCI and CASE. Phil Sully, Consultant Manager, CGI-LS3.

10.00 Coffee

10.30 Task oriented graphical systems. Paul Seaton, Consultant, System Concepts.

11.15 The PC of the 1990s: the move toward computer supported collaboration. Ian Wilson, UK Technical Marketing Manager, Intel Corporation.

12.00 LUNCH AND EXHIBITION VISIT

C++

Chairman - Mike Banahan, Chairman, European C++ User Group.

14.00 Experiences of migration from C to C++. Kerry Williams, Software Specialist, Swiss Bank Corporation.

14.45 Selecting class libraries. Dave Mansell, Managing Director, Citadel Software.

15.30 Tea

16.00 Lessons learned on C++ projects. Keith Laidlaw, Database Specialist, British Airways.

16.45 Practical experience of C++ using prototyping. Kevin Large, Principal Consultant, Hoskyns Group.

17.30 Close

TRACK 2

AM: WINDOWS DESIGN AND ANALYSIS

PM: WINDOWS PROGRAMMING

Chairman - Steve Baker, QA Training.

QA Training provide technical training, consultancy and development tools in most programming technology areas. QA have been working with Windows for many years and are Microsoft's Systems Partners.

Steve Baker is a well known speaker, having written on development issues and is currently a technical adviser to the Windows Development Association.

Developing Windows applications is a high priority for many companies. These sessions will focus on key Windows development issues, features, leading Windows developers and industry experts.

Aimed at practising developers and development managers, the sessions will be full of practical, technical advice and information from experienced Windows developers to newcomers and existing users.

The sessions cover topics such as:

- Technical directions in Windows 16 bit and 32 bit Windows development
- Key Windows development tools C & C++ compilers, toolkits and '4GL' tools
- Object oriented development and windows
- Multimedia applications
- C++

INSTITUTE OF Software ENGINEERING

QA TRAINING

CONFERENCE REGISTRATION FORM

HOW TO REGISTER: Please photocopy and complete sections A-D on the form below to apply for your place, and return it either:

BY POST TO: Conference Registrations Department, Blenheim Online, 630 Chiswick High Road, London, W4 5BG.

BY FAX: +44 (0)81 742 3286 or +44 (0)81 747 3856

ENQUIRIES: For Conference Registration enquiries please telephone Jackie Wood, all other enquiries please telephone Lynne Davey Tel: +44 (0)81 742 2828

(A) PLEASE TICK WHICH SESSIONS YOU WOULD LIKE TO ATTEND:

DATE	TECHNICAL CONFERENCE	
	Track 1	Track 2
16 June AM	OPENING PLENARY – SETTING THE SCENE. By The Institute of Software Engineering ☐	
16 June PM	Database architectures & programming ☐	UNIX ☐
17 June AM	Networks architecture and operating systems ☐	OOPS design ☐
17 June PM	Expert systems ☐	OOPS programming ☐
18 June AM	Human computer interface ☐	Windows – design and analysis ☐
18 June PM	C++ ☐	Windows – programming ☐

(B) BOOKING OPTIONS

1. CONFERENCE AND EXHIBITION

Fees include:
- Free entry to the exhibition
- Full conference documentation
- Use of VIP Lounge in the exhibition
- Free exhibition show planner
- Lunch and refreshments

I will be attending: Amount due

☐ Opening plenary @ £90.00 + VAT = £105.75 £.....

☐ ½ day technical sessions @ £90.00 + VAT = £105.75 x _____* £.....

Passports – enable delegates to attend all sessions throughout each programme.

☐ Technical Passport @ £450.00 + VAT = £528.75 – a saving of £90.00 – attend one session free! £.....

* Please indicate how many sessions you are attending. **Total £.....**

2. EXHIBITION ONLY

Register now and save £10.

☐ I am not able to attend the conference but would like to pre-register for the exhibition.

(C) PERSONAL DETAILS (Block capitals please)

FULL NAME (DR/MR/MRS/MS) _____

JOB TITLE/POSITION _____

ORGANISATION _____

ADDRESS _____

_____

POSTCODE/ZIP _____ COUNTRY _____

PHONE _____ FAX _____

PRINCIPAL ACTIVITY OF YOUR ORGANISATION _____

SIGNATURE _____

(D) METHODS OF PAYMENT (Payment must be made prior to the conference)

1. Please charge to my credit card:

☐ American Express ☐ Visa ☐ Mastercard/Access ☐ Eurocard

Card number | | | | | | | | | | | | | | | | |

Expiry date _____ Name (as it appears on card) _____

Card billing address _____

_____ Signature _____

2. ☐ Fee enclosed - cheques payable to Blenheim Exhibitions and Conferences Limited

3. ☐ By bank transfer to Royal Bank of Scotland, London Western Branch, Account number 00120387, Sort code 16-01-29

Transfer date _____

4. ☐ Please invoice my organisation.This option is only available until 2 June 1992 to allow time for processing. Bookings received after this date must be accompanied by payment. All invoices must be paid prior to the event.

Confirmation and a VAT receipted invoice will be forwarded within 10 days.

BLENHEIM ONLINE

Blenheim Online, Blenheim House, 630 Chiswick High Road, London W4 5BG.
Tel: 081 742 2828 Fax: 081 747 3856/4012

Blenheim Online is a division of Blenheim Exhibitions and Conferences Ltd.

CIRCLE NO. 920

AD/EXE

the assignment attempt (the ability to force a type on a variable) which is absolutely essential; and there are the persistence facilities. Going a little bit beyond the language, the presence of standard libraries is, I think, a really strong plus for the Eiffel approach. In C++ there are all kinds of competing libraries but nothing has emerged as a real standard, in part because the language does not support the tools (such as genericity) for building libraries.

Template faults

With regard to genericity, what is wrong with templates as implemented in C++?

To start off with, templates are only now getting into the language. But they are only an emulation of genericity. In particular they are not closely connected to the type system, and there is no support for constrained genericity. It is very important to be able to accept actual generic parameters only if they are descendants of a certain class. For example: I want to have vectors of something of type `T`, and I want to be able to add two vectors, so objects of type `T` must have the + operation applicable to them. It means you can then have a vector of integers, but you cannot have a vector of nuclear submarines if a nuclear submarine doesn't have a + operation.

Templates cannot support this because they are just a kind of macro. I see templates as making more official what people had been doing manually in C++ so far, which is using the pre-processor to generate variations of a class.

But templates do give you some type protection... You're saying that's not enough?

Yes, I am. Type protection in C++ is in any case always problematical. As long as you can have those casts, as long as you can convert from any pointer type to any other pointer type, what type we talk about becomes pretty meaningless.

In a recent interview with .EXE, Bjarne Stroustrup said that he didn't think that it was necessary to add Eiffel-style support for assertions to C++. He thought it sufficient that one could acquire add-on specialist tools to do the job, and that he did not believe in cramming too many features into a language. Is that not a fair comment: if you need assertions, you should buy a separate tool?

The comment 'I don't believe in cramming too many features into a language' is a fair comment. I would say that about C++. I don't think it is proper to have, for example, function pointers and dynamic binding in the same language, because it is confusing. If you don't believe in crowding too many features into a language, then you wouldn't produce C++, because C++ is exactly that. It's taking C, which already had its share of language features, and adding more, including some which are redundant and incompatible.

This business of function pointers is typical. If you use an object-oriented language like Smalltalk or Eiffel, then to obtain automatic selection from various operations at run-time you use dynamic binding. If you use C, you can emulate this in a rather unpleasant low-level fashion by having arrays (or other data structures) of function pointers. Now this is another way of doing things; it's less nice, but it works. What I don't think is proper is the C++ approach in which you have both mechanisms. Programmers have to choose all the time between the one way of doing things and the other, which means a lot of confusion and complexity.

I decided that I should learn up about C++ after all, so I went and read the Ellis and Stroustrup book *(C++ Annotated Reference Manual)* from cover to cover. I was horrified to see how many criticisms there are in that book of C and C++. There are comments like, 'the array facility of C, and hence of C++, is brain-damaged.' You read this and you say, 'What? You're designing the language, and now your telling us that something is still wrong?'

There are comments like this on many other aspects. There are comments like, 'This is available, but don't use it.' I don't think that is good language design. If you produce a language design book, you should be proud of it and there shouldn't be any dark corners. I think I can say that about Eiffel. I'm not saying that Eiffel is perfect, but I cannot point to any construct in Eiffel for which there is a comment of the form 'Don't use it'.

Returning to your question: I don't think that assertions should be separate from the language. Assertions are absolutely essential to object-oriented design. This was something which was mentioned in *Object-oriented Software Construction*, but clearly

A Few Eiffel Terms

Assertions - C programmers will be familiar with the macros in ASSERT.H, which allow the programmer to assert that a condition is true. If it is not, the program aborts with an error message. An example use is to protect a routine that calculates square roots of positive reals from being called with a negative parameter.

Although Eiffel's assertions are superficially similar, they are fully integrated into the language (for example, they have their own set of inheritance rules) and are much more powerful. There are various kinds. *Preconditions* work like the square root example above: 'before you call this routine, *this* condition must be true.' *Postconditions* say what must have happened on the completion of a routine. A simple example is a routine which adds an item to a list; its postcondition might state (among other things) that the list is non-empty on return. *Invariants* are attached to classes rather than routines; these make statements about a class's data that are always true. For example, if the list class has a Boolean field `empty` and an integer field `count_of_items`, then a reasonable invariant is `empty = (count_of_items = 0)`.

As well as their debugging use, assertions are used in documentation of classes, and to enforce 'Design by Contract'.

Assignment attempt - this is Eiffel's mechanism for identifying object types at run-time. Using a special assignment symbol `?=`, the programmer is allowed to attempt to assign to an known object from one of unknown type (perhaps unknown because it has just been loaded from a database). If the type system prohibits the assignment, the target object becomes void; if it is allowable, a normal assignment takes place. This system fulfils the same purpose as Turbo Pascal's `TypeOf` function and C++'s proposed `ptr_cast()` and `ref_cast()` 'safe cast' extensions.

Genericity - Generic classes are 'typeless' container classes which are assigned a type when instantiated. For example, one could build an all-purpose `list` type, then use it to create a list of numbers, a list of windows, a list of addresses. Please see Meyer's comments in the text for an explanation of *constrained genericity.*

Type-checking - Eiffel is a very strongly typed language. In particular, *all* type-casting (which Meyer describes as 'a sordid back-alley deal') is forbidden.

not enough. I've written more about it since then. It's this whole idea of Design By Contract. That, for me, gave the theory behind object-oriented software construction. It's not just that you have classes and inheritance and so on; it's that you build software in such a way that it's made of pieces that communicate with each other on a basis of well-defined obligations and benefits.

I have tried to explain Design by Contract in a chapter with that title, part of a collective book that has just appeared, *Advances in Object-Oriented Software Engineering*. It's this idea that software is a combination of various pieces which communicate with each other not on the basis of pre-defined assumptions, but on the basis of proper and precise definitions of what each one of them expects from the other and must provide. This is what justifies the idea of preconditions, postconditions and invariants, and I don't know how to teach object-oriented programming without them. When I teach object-oriented techniques, I spend anywhere between one third and one half of the presentation on assertions - especially in connection with inheritance. I don't think anyone can understand inheritance properly without introducing assertions.

I also don't think you can understand the notion of class without the idea of the invariant, which expresses the integrity constraints on a certain data type independently of how the data type is implemented. This is where I disagree with Bjarne Stroustrup. I don't think assertions are 'fairly useful'.

Be assertive

Do you think people who use Eiffel always use the assertion mechanism?

People who use Eiffel well use them a lot. Even people who don't include assertions in their own software, because they haven't yet understood the power of this notion, benefit from them anyway.

The practice of software development in Eiffel is pretty much based on libraries. When you switch to Eiffel, you don't necessarily see as the major change a change in language or in method; what you see is a way to start working at a higher level of abstraction by using libraries. Now, these basic libraries are fully loaded with assertions. The documentation is essentially based on assertions, and their use is based mainly on assertions too. So even if somebody is only starting to work with Eiffel, and has is not yet putting assertions in his own software, he is going to benefit from them anyway.

It is true, however, as you implicitly suggested, that some people who start with Eiffel - especially if they come from something else, like C++ - don't necessarily put assertions to their full use. But they usually start using them after some experience.

Another point about assertions: when reading your book** Object-Oriented Software Construction **I found that it was not always obvious to me, as a programmer rather than a mathematician, why particular assertions were applied - particularly in the case of the invariant. Is it just that I am not smart enough, or have you found this to be a more general difficulty?

In my experience, and the experience of people working with me - both developers in our company and users - it is true that you don't necessarily get the invariant right first time. As you start improving a class, adding things to it, and understanding it better, this process is pretty much embodied in writing the invariant and improving the invariant. The more you understand what a class is

about, the more you are able to express invariant clauses and; when you add a clause to your invariant, then you gain something in understanding what the class is about.

This doesn't necessarily come right the first time - but proper software design doesn't come right first time anyway. The process of improving the software and the process of writing and improving the invariant go together. It's not surprising that these things should appear a little difficult at first - software design *is* difficult. But invariants, and assertions in general, enable you to get to the heart of the matter.

Garbage

Can we talk about memory management. Why do you use garbage collection, instead of 'manual' heap managing systems?

In the manual for Eiffel-S, which is the SIG implementation for DOS, they have a very nice analogy. They say something like: 'An object-oriented program without a garbage collector is like a pressure cooker without a valve. You don't know exactly when it is going to happen, but you know that sooner or later it is going to explode'.

An object-oriented program generally creates a lot of objects. A lot of them are going to become unreachable. You can of course make sure that not too much of that happens - having a garbage collector is not an excuse for generating tons and tons of garbage, you still have to be careful - but if you start managing these things yourself it's dangerous and it's tedious.

It's dangerous because you always run the risk of 'freeing' (in the sense of C's `free()`) an object which in fact is still needed. This is the source of some of the worst bugs that exist in C programming. It's a very serious problem, because the consequence of the error is usually quite remote from the source of the error. Usually when you think you can free an object and you are wrong - there is still some reference pointing to it - you use that reference much later in the program, so that tracing back the cause of the error may be extremely difficult.

As for the tedious part: if you do your storage management manually, you end up polluting your code. If you want to do manual reclamation, you have to write a lot of recursive `free` procedures. As long as you have the kind of complicated data structures that are possible in object-oriented programming, it's not enough to free one object; you have to follow the pointers. You end up being the garbage collector yourself. Programmers have better things to do with their time.

The Ellis and Stroustrup book is the most damaging criticism of C++ that I know

It seems to me that if you don't have a garbage collector, you lose many of the major benefits of object-orientation. Personally I wouldn't write an object-oriented program in an environment in which I didn't have a garbage collector.

C programmers are hostile to garbage collectors because of the time overhead...

That's just because they don't know about modern garbage collection technology. In version 3 of our implementation, we estimate that the overhead of garbage collection - the difference between running a program without garbage collection or running it with garbage collection - is about 20%. But this is not the real overhead, because if you didn't have the garbage collector, you would have some overhead due to manual reclamation anyway. With version 3 - version 2.3 was not as good in this respect - we do not expect anybody to run an Eiffel program without garbage collection enabled. But you can still switch it off if you want.

On the street

How many implementations of Eiffel are there at the moment?

Among the ones I know about are: there's of course ours - Interactive Software Engineering's implementation of Eiffel; there's an implementation by SIG Computers of Germany, which is essentially a DOS and OS/2 implementation, although there's also a UNIX version of it; there's a company in the US called Power Solutions that is about to release its implementation; and there's a GNU version that is in the making but is not released yet. Oh, and I hear that there's some people called Nexnix Ltd in Brighton who are developing a compiler.

By the way, the ISE implementation will support DOS/Windows some time later this year.

What are the main features of version 3 of the Eiffel language and environment?

Probably the most important thing is this 'melting ice' idea, which is an attempt to get the best of both the interpreted world and the compiled world. If you look at why some people use environments such as LISP systems and Smalltalk systems, once you have removed the superfluous arguments, it boils down to just one, practical, serious idea. With an incremental environment such as Smalltalk - I want to say 'interpretative environment' but that's not quite accurate - you can get a very quick turn-around.

Until now, with statically typed, compiled, object-oriented languages you have had to go through a fairly classical edit-compile-link-execute cycle. This means in a big system, even for a small change, there's a fairly long wait. This is an issue we have been grappling with for a long time, because there is absolutely no philosophical reason why you should have to choose between a quick turn-around and static typing.

Static typing is good. It gives you safety, because you are able to catch errors much earlier, and it gives you efficiency, because it make it possible to generate much, much better code. And there's no reason why these goals should be incompatible with a very, very quick turn-around.

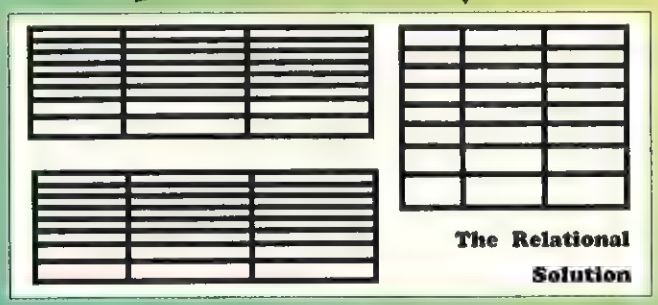
This is essentially what this melting-ice technology of Version 3 achieves. You can have extremely quick re-execution after a change, even though you retain the static typing. The idea is very simple. Whatever you change inside the normal development cycle is going to be interpreted, so that it's extremely fast to see the results of a change after you have made it. The interpreter doesn't have any significant negative effect on performance, because typically you're dealing with a big system and after a change only a small part of it will have been affected, so most of it will still be compiled. We think that this is a major advantage that will make all the difference in the world.

Another aspect to version 3 is the presence of the library called EiffelVision. This is a graphical user-interface library supporting various tool kits - to start with Open Look and Motif - with complete source code compatibility. So people can write the best

Does the Real World fit your Database, Or your Database fit the Real World ?

Problem :

To Efficiently Store Complex Real World Objects in a Database.



"Announcing P.O.E.T, the first true Object Oriented Database system for DOS, Windows, UNIX and NeXT, with prices starting from just £249"

The Affordable Database Solution.

Up until now, object oriented DBMSs have typically only been available on high end UNIX machines, with prohibitive entry prices of $20,000++. Now you can unleash the power and potential of a true OODBMS in your DOS or Windows application, at prices starting at £249 +VAT.

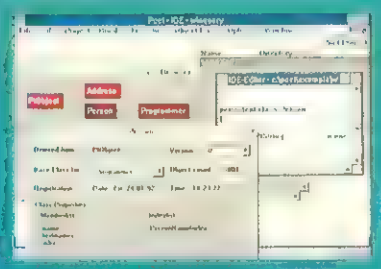
Object Oriented DBMS v Relational

Relational Database (RDBS) and B-Tree/ISAM systems require that data be portrayed as a series of two dimensional tables. These systems were a real breakthrough in their time, but software developers are rapidly learning that life is not a series of two dimensional tables.

For example, a CAD program might have an array of shapes, or a DTP program might model a page as a series of frames which might contain bitmaps, paragraphs or vector drawings. Conventional databases have a fixed set of data types. And the only way they group data is in a table.

Object-oriented programs allow us to declare new data types as needed. And POET stores the objects we create ***as*** objects. This one-to-one correspondence of the database design to its physical storage can realize access gains of over one hundred fold compared with relational DBMSs. POET has the speed while keeping its kernel small; the DOS runtime is under 256Kb.

POET includes complete support for C++ inheritance and encapsulation, class dictionary with class versioning, optimized indexing and querying, object caching, complex operators and containers. POET has what it takes.



Persistent Object Extended database Technology

POET took over 20 man years to develop by leading German Software house BKS. In non German speaking Europe it is distributed by Silicon River Limited, the same team that first popularized C and C++ in the UK by aggressively pricing Zortech C/C++.

What Should You Do?

This is the powerful, flexible database technology you have been awaiting. Now it is not only available, but affordable. If you will be developing a database application on any of the platforms below, then please call for a free Technical Overview of POET, the object oriented DBMS engine for the 1990s.

Platforms Supported:

Borland C++ (DOS or Windows)
Microsoft C/C++ 7.0 (DOS or Windows)
Zortech C++ 3.0 (DOS or Windows)

LPI C++ (UNIX 386 Systems)
HP C++ Softbench
GNU C++ (NeXT Step)

For your free POET Technical Overview call

081 317 7777

Silicon River

Silicon River Ltd 58-60 Beresford Street, LONDON, SE18 6BG
Voice 081 317 7777 Fax 081 316 7778 BBS 081 317 2310

CIRCLE NO. 922

into user interface applications in terms of high-level concepts like menus, windows and so on, without being concerned with the details of Motif etc. Then they can just port their application to various graphical platforms without making source changes. Eventually this will also apply to Windows and Presentation Manager.

Also important in version 3 is the availability of the standardised relational database interface. This follows the same principles as our GUI library; that is to say you program in terms of the SQL model and then you go to some other RDBMS without source code changes. This is very important for big projects in commercial areas. Most big relational systems currently use C, so it is very important to be able to access traditional data quite easily and to map objects onto relational data.

Looking forward

A recent .EXE Survey showed that about half our readers used C or C++ as their primary development language, about 10% use Pascal, and no other single language collected more than about 5%. There were no Eiffel users. What is your prediction for the result of next year's survey?

Let me consider two years from now, when Eiffel will have had a chance to make an impact on the DOS world. I think that people coming from languages like Pascal and Modula-2 will very naturally migrate to Eiffel. They'll find themselves on safe ground with strong typing, general software engineering concerns and so on. I would see them migrating *en masse* to Eiffel.

I think a proportion of people programming in C will migrate to Eiffel as well. Those are the people who, again, have serious software engineering concerns, and want to be able to guarantee the quality of the code they produce. I don't know how much that is - it's a certain proportion of people writing in C today.

As for people using C++ today; they're still to a large extent the *avant garde*, the early pioneers. I would say that once these people have understood the benefits of object orientation and the limitations of C++, they will look for something more serious. I don't think there is much competition to Eiffel.

So without making any too wild predictions, I think that, if you repeat your survey two years from now, from 15% to 25% will be using Eiffel.

And C++?

I think that five years from now, no-one will be using C++.

This despite the fact that 80% of our C users are looking at C++?

That's all they have heard about. To a certain extent you cannot expect anything else. Suppose somebody has been raised to the Stalinist creed, who has only ever read political literature favouring Marxism and Leninsm. If you ask him, 'What is the next thing?', he's not going to answer 'Democracy and the free market.'

EXE

Many thanks to Dr Meyer for taking the time to give this interview. To find out more about Eiffel, you should contact Caroline Browne at Applied Logic Distribution (081 780 2324).

With the Interface Builder,™ you can use the mouse not only to place and resize your objects, but to change their attributes and define interaction – all without writing code.

For even greater power, the NeXTstep object library is wide open. That is, you can customise any object's behaviour through the use of subclassing or create new objects.

Because you can reuse objects, there's no need to reinvent the wheel with every new project. And to update, you need only to modify a single object.

With NeXT workstations, custom software is both intuitive and consistent, offering stunning graphics and access to all system services.

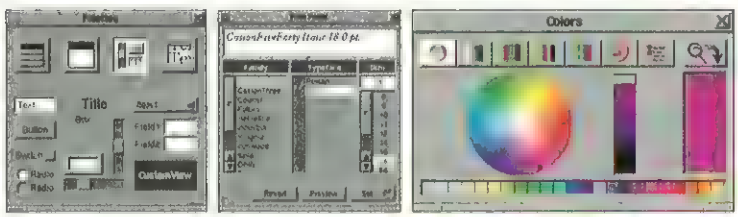
Further, it integrates seamlessly with other applications, custom and commercial.

For more information on NeXT computers and our developer programme, please complete the coupon or telephone 081 565 0005.

CONGRATULATIONS. YOUR NEXT CUSTOM APPLICATION IS ALREADY HALF FINISHED.

The graphical revolution made computers far easier to use, but infinitely more difficult to program. That's why the industry is so intent on object-oriented systems, where powerful applications are created in a fraction of the time.

For most companies, just the first effort is still at least three years away. But NeXT™ has a proven system that works today.



The NeXTstep development environment delivers all the benefits of a true object-oriented system. Sophisticated application objects are all provided, along with the power to subclass and create exactly the objects you need.

The NeXTstep™ environment is an object-oriented world right down to the system level, offering tools you've never had before.

NeXTstep's Application Kit™ gives you the objects common to most programs, ready to use without any modification. It even includes objects for faxing, choosing fonts/colours and interapplication communication – all with standard dialogues.

I'd like to receive more information on NeXT ☐

Please complete and return to: NeXT, 1 Heathrow Boulevard, 286 Bath Road, West Drayton, Middlesex UB7 0DQ.

Name:

Company

Address:

County:

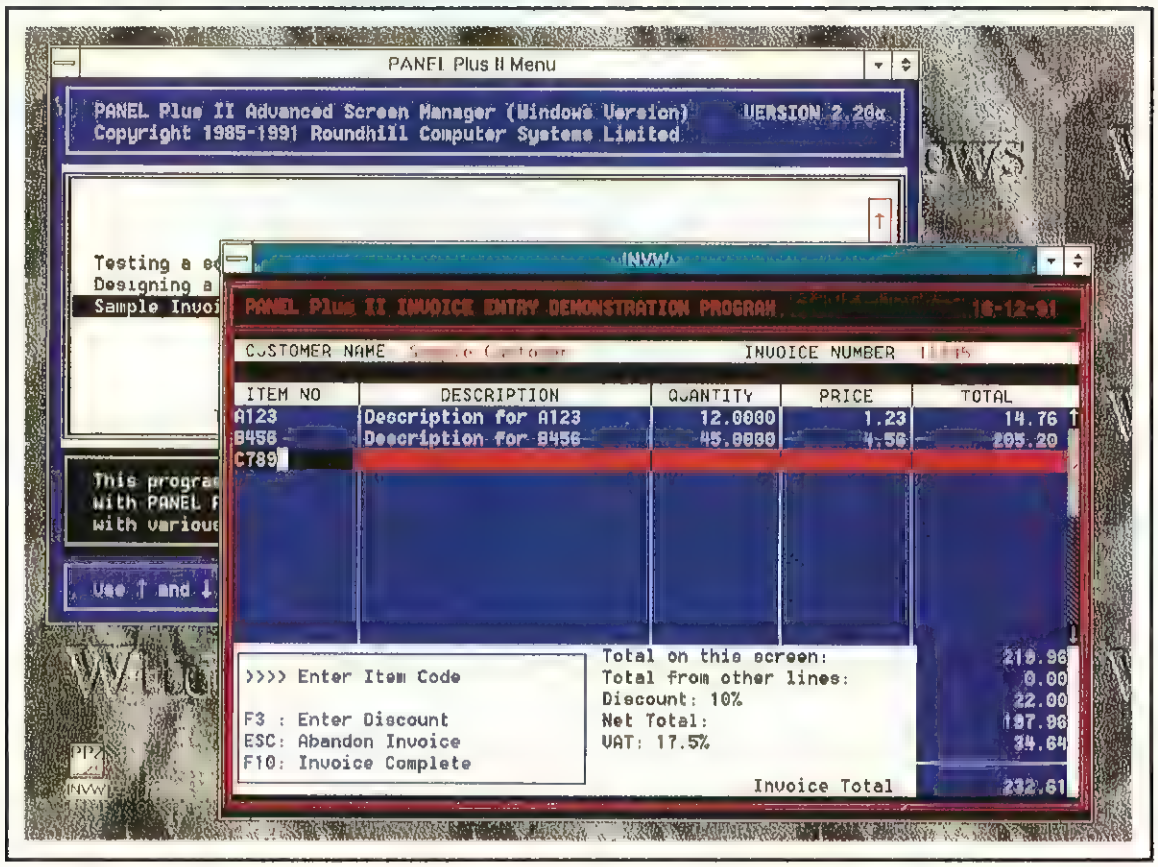
Telephone: Post code:

.EXE 5/92

CIRCLE NO. 924

PANEL® *Plus* II

Screen Generator and Screen Management Library for 'C' under DOS, Windows, OS/2, Unix, Xenix, QNX, VMS or RMX



Release 2.20 of Roundhill's **PANEL *Plus* II** user interface manager adds MS-Windows 3.x to the list of supported environments. Applications created using **PANEL *Plus* II** are source-level compatible across all platforms. **PANEL *Plus* II** includes a powerful and easy-to-use screen design utility, which allows specification of all field attributes, including built-in and custom validation, scrolling regions, and menus in many different styles. Editing and field-navigation keys can optionally be specified as CUA/SAA compatible. Screen layouts can be loaded dynamically by your application, either from a disk file or a Windows resource area, or the **PANEL *Plus* II** C code generator can create field definition structures and data areas for inclusion in your program.

The DOS-Windows-OS/2 version of **PANEL *Plus* II** includes pre-built libraries and DLLs for all major C and C++ compilers (and DOS-extenders). Royalty-free library source included with all versions has low-level specific files for all supported operating systems and compilers.

The **PANEL *Plus* II** standard licence at £285.00 per programmer includes documentation, sample programs and tutorials, libraries and DLLs, library source code and header files. Source for the screen editor and code generator is also available for self-porting to new C environments. Please call for a full version and price list and a free demonstration diskette.

Roundhill Computer Systems Limited

Orchard House • Ogbourne St. George • Marlborough • Wiltshire SN8 1SU • England

Fax: 0672 84 525 • Tel: **0672 84 535**

CIRCLE NO. 925

New Improved Lint

The C language has so many pitfalls for the unwary that it is a wonder programs ever work. As Dave Stiles discovers, Gimpel Software's Lint is a major aid to ensuring they do.

C Compilers are not too hot on reporting the more subtle programming mistakes. They are usually quite happy to accept a reference to an uninitialised variable or an assignment of an over-large constant; and if a `switch case` is allowed to flow into the next one, fine.

This is reasonable behaviour for a compiler. It has enough trouble looking for real errors, without checking programming logic. Hence the existence of Lint.

Lint? It picks up all the fluff from your source code, leaving a nice, clean program. It gives a much more in-depth analysis than a compiler ever could. It proceeds through one or more source files, in one pass, and tells you what is wrong with the code.

Lint also reports potential portability problems and pin-points non-ANSI and other suspicious coding practices. It ends by generating a wrap-up report which details unused declarations, definitions and header inclusions, undeclared variables and functions and so on. When several files are processed, a global wrap-up is generated. This shows functions, variables etc which are unused in the files and indicates global variables and functions which could be local to a single module.

Open the Box

Gimpel Software's PC-Lint V5.0 (Lint V5) is supplied in the usual form of three-ring binder, shrink-wrapped manual and disk package. Both disk formats are included: 720 KB and 1.2 MB. It operates on any PC, XT, AT or PS/2 compatible under MS-DOS 2.0 or later. It also works under Windows and in an OS/2 DOS box. (Gimpel also markets a generic lint, under the title FlexeLint, for use on all other platforms.)

Lint V5 assumes your C source code is to K&R and ANSI standards; where there is conflict, ANSI overrides K&R. As it was completed after the ANSI Committee closed up shop, it should meet the standard.

The 300 page manual is reasonably well laid out and easy to use. It includes sections on installation and configuration, special features and language extensions, and preprocessor handling. A major section is devoted to *strong types*, a new feature introduced by Gimpel.

The largest section is devoted to the messages; I estimate there are about 400, of which 80 are new to V5. There are four types of consumer messages (errors, warnings, information and elective notes) and two varieties of run-time errors. Messages are explained in full, usually with an indication of what to do about them.

One section gives differences between V4 and V5; I wish all manuals did this. For those upgrading, this is probably the first section to read. A section for newcomers is Living with Lint: a quick guide to overcoming bad programming habits without too many tears.

I opened the disk package with trepidation, as it claimed to be for 386 processors. Mine is a 286, and I'd been had before. I put the disk in, typed A:INSTALL... and watched the computer die.

Lint V5 includes the Phar Lap DOS extender. This does not normally crash a machine, even if it cannot run. The Install program was not Phar Lap's (nor was it Gimpel's), and it made a mistake. It assumed that expanded memory was to a recent EMS standard. My machine, a rather old AT clone, has no address line 20 and only a primitive EMS Version 3.0. So, legs in the air.

```
//    Compiler Options for Microsoft C Version 6.0 and earlier
-cmsc
      This file contains options to allow PC-lint to process source
/     files for your compiler.  It is used as follows:
'
 /    lint  co-msc.lnt  source-file(s)

        // If you're programming MS Windows set the option
        .    +e934 to report passing short auto pointers as arguments
        // See Petzold, Windows 3.0, pp 897-902
           You might also want to set the option: -printf(w2,wsprintf)
        // It makes wsprintf like sprintf except that pointers have
        // to be far.
-format=%(%f(%l)\s:\s%)%t\s%n: .s%m
        // error format expected by Microsoft editor
        // Note that %c can also be used to specify column
-e46    // allows bit-fields to be other than int or unsigned
        // If you are use Microsoft Editor try using -hfa 3
        // the 'f' tags most messages with a location.
        // To tag ALL messages with a location use -hF...
-elib(514)  // allow #if <boolean> | <boolean>
+fan    // allow anonymous unions
+fdi    // Use directory of the including file
+fcd    // makes cdecl significant -- needed for MSC option -Gr
-esym(123,min,max)  // allows users to use as variables
 d_MSC_VER=600  // The version number of Microsoft C (alter if necessary)

          // while processing compiler (library) header files ...
-elib(652)  // suppress message about #define of earlier declared symbols
 elib(762)  // suppress message about multiple identical declarations and
-elib(760)  // suppress message about multiple identical macro defs

/,  The following functions exhibit variable return modes.
//  That is, they may equally-usefully be called for a value
//  as called just for their effects.  Accordingly we inhibit
/   Warning 534 for these functions.
/,  Feel free to add to or subtract from this list.

-esym(534,close,creat,fclose,fflush,_flsbuf,fprintf,fputc)
-esym(534,fputs,fscanf,fseek,fwrite,lseek,memcpy,memmove,memset)
-esym(534,printf,puts,scanf,sprintf,sscanf,strcat,strcpy)
 esym(534,strncat,strncpy,unlink,write)
```

Figure 1 - Compiler Options Script for Microsoft C

PC-Lint itself, when finally installed, was less brutal: it merely told me it could not run without a 386 and left the machine running.

I transferred to a bare bones AT which had no extra memory, and Lint installed with no problems. The session consisted of simple questions, with options to install for a primary compiler and as many others as I wished. As with all polite programs, it asked for the target drive and directory.

Two variations of Lint V5 are supplied: LINT.EXE, compiled for a 386 processor and using the Phar Lap extender; and a bound DOS-OS/2 version L.EXE. The distribution disk includes two utilities to aid in setting up the Phar Lap extender. Total installation size is about 550 KB; if you remove the unused version, this drops by 200 KB or 350 KB.

The installation includes a mere handful of short example files. I guess the idea is that your own code will provide hours more amusement and education than any example code ever could.

Configuration

Since I selected a Microsoft compiler, the installation program installed compiler definitions file CO-MSC.LNT (Figure 1). Lint V5 comes with 28 preset compiler definitions, from the common to the obscure. Among the latter are Intel 8051 and 8196/8198 compilers. If your compiler is not included, the generic file CO.LNT can be tailored to suit.

Gimpel also supply a generic library header, SL.C, which you can tailor if your compiler does not include function prototypes. Five preset headers are also supplied.

Lint is normally run from a batch file, to simplify control of command line switches. Output is directed to a file. This, believe me, is necessary; there can be an awful lot of messages. I send my output to RAM disk and include a line in the batch file to load the error file into an editor.

Lint's command line comprises a mix of option switches and file names. Anything without a switch prefix character (+,-) is assumed to be a file name. If the name has no extension, PC-Lint looks for a matching file with an extension of .LNT, .C or .H before giving up. Wildcards are acceptable: LIN *.C goes through all the .C files in the current directory.

The command line option `-iC:\LINT` tells PC-Lint to look for .LNT files in directory C:\LINT; similar to the DOS APPEND command.

You can specify any number of .LNT control files or .C source files on the command line or in control files. The original installed system specifies only a single control file, STD.LNT. This contains a reference to OPTIONS.LNT and to the compiler definition file (eg CO-MSC.LNT).

OPTIONS.LNT contains parameters which you decide are best for your way of working. Where these vary between projects, place a copy of the file in your working directory. A typical reason for this would be to include various compile-time switches and macros; eg `-DDEBUG_ON`. If this is not done, code defined inside a `#if` clause will be ignored by lint.

Besides placing lint options inside .LNT files and on the command line, they can also be included directly in source files. If the word *lint* follows an opening comment, the text following it is processed by lint. For example `/*lint -esym(715, mgn, mgs)*/` instructs lint to ignore warning 715 when it applies to variables mgn and mgs.

Any warning, error or information message can be turned on or off globally, or suspended locally and then reinstated. This can be carried out with `esym` for named items only, or for all items (eg -e715). Where necessary, whole groups can be cancelled (eg -e7??). However, this is obviously counter-productive if taken to excess.

The First Test

I passed a module from my latest program, validated under Lint V4 with a few option overrides, through V5. I specified only the default options, to see what would happen. Lint threw up 12 errors and 15 warnings. Of these, nine errors were due to signed/unsigned mixes in constant strings, of which I was already aware.

Another error concerned a mix-up over an `int` pointer; an unsigned pointer went into a function which wanted a signed one. The variable was only used as a boolean flag, so it did not actually matter. However, I will now reconsider my policy of turning off signed/unsigned pointer checking.

The other errors involved a re-declaration of functions between the header, where they were declared as `int _cdecl myfun(void)` and the source file, which had `int myfun(void)`. I declared these prototypes under Lint V3. V4 accepted them. However, V5 option `+fcd`, in CO-MSC.LNT, specifies that `_cdecl` is significant. Since the function definitions in the source code did not include `_cdecl`, Lint complained.

The warnings generated on my code all complained of an inconsistent return mode from called functions. A return from a C function can be used or ignored. In many cases, eg `strcpy()`, it does not matter; however, functions such as `atoi()` would be useless if the returned value were ignored.

PC-Lint allows you to specify functions as having an ambiguous return mode. This is

```
enum { PAG_ASSLST, PAG_PERLSTR };
int fldid;

  switch(prvpag)
    {
    case PAG_ASSLST  : fldid=QFP_CNAME; break;
    case PAG_PERLSTR : fldid=QFN_RECID; break;
    }
  load_name(fldid);

Lint comment:
     load_name(fldid);
edt.c(161) : Warning 644: fldid (line 154) may not have been initialized
```

Figure 2 - Possibly Uninitialised Variable

```
#define F12 (unsigned int)(134<<8)

Lint comment:
    case F12 : do_f12(); break;
edt.c(509) : Warning 648: Overflow in computing constant for operation:
             'shift left'

Repair:
#define F12 (unsigned int)((unsigned int)134<<8)
```

Figure 3 - Constant Overflow

```
    case K_QUIT : exit(0); break; /* exit() unrecognised as end by Lint */
edt.c(1255) : Warning 527: Unreachable
```

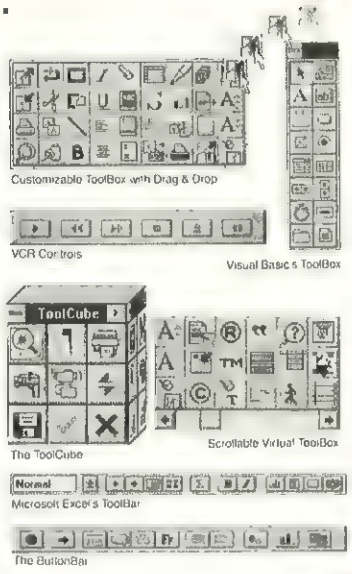
Figure 4 - Exit as End of Switch Case

The Right Tools for the Right Job

TbxSHIELD

The fastest way to develop toolbox controls like . . .

- Dynamic Link Library
- Works with Visual Basic, ToolBook, Actor, Borland C++, SDK & more . . .
- Microsoft Windows or OS/2 PM Version



TbxSHIELD for Windows **£225**
OS/2 **£225**

LIANT C-scape®

The latest release of C-scape — the best selling character/graphics based screen interface C library in the UK. Runtime royalty free. Includes Look and Feel™ Screen Designer and Source Code. Available for DOS, most DOS Extenders, UNIX, X Windows and VMS.

DOS — **£370** (incl source and LnF) UNIX — **£1065** (incl source and LnF)

Heron Software C-scape Extension Library

New version 2		Price
Enhanced set of functions for C-scape 3	DOS	**£160**
Includes Source Code.....	Unix	**£250**

Phar Lap 386|DOS Extender™

Beat the 640K DOS limit with a flat 32-bit address space on 386 processors. Supports a wide range of 32-bit languages. SDK **£330**

Phar Lap 286|DOS Extender™

C beyond 640K with your Borland C++, Microsoft C or MS Fortran compiler on any 80286, 386 or 486 platform. SDK **£330**

MagnaCharter II

Enhanced version of the much acclaimed MagnaCharter charting package for Windows 3, 99x99 cells with wide range of printer support. Complete set of symbols — or add your own..... **£160**

Prices are exclusive of VAT and subject to change.

Orders and enquiries to:

SYSTEMSTAR LIMITED

1-3 Parliament Square, Hertford, SG14 1EX
Telephone: (0992) 500919 Facsimile: (0992) 554261

CIRCLE NO. 926

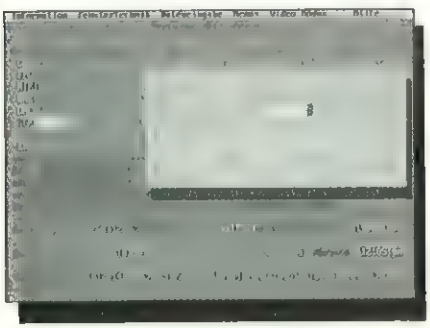
Your interface doesn't have to be a straightjacket

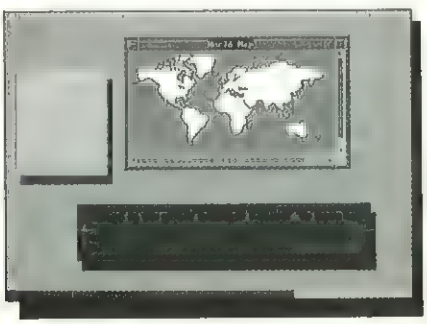
C-scape® is the most flexible user interface package for text or graphics based applications.

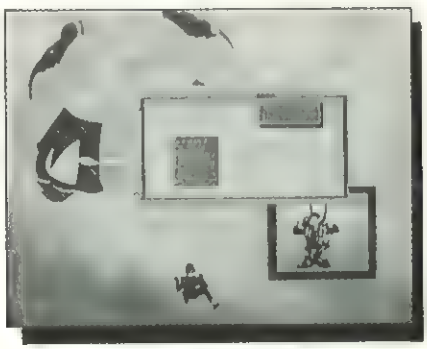
C-scape sets you free with powerful features such as scrolling windows, mouse support, menus, text editor, help, data entry etc and lets you add your own functionality to these objects without touching the underlying code.

C-scape's object oriented architecture makes it easy to integrate with any other C library and applications port without modification across DOS, extended DOS, OS/2, UNIX, AIX, X Windows, QNX and VMS.

C-scape's Look and Feel® interactive screen designer lets you create full featured screens, test them and then automatically generate complete C source code or binary screen files.







An open systems tool from Liant Software distributed and supported in the UK by Systemstar, C-scape includes source code and is run-time royalty free.

Free your creative talent and see for yourself why thousands of programmers worldwide use C-scape. Call Systemstar for more details.

LIANT

SYSTEMSTAR LIMITED

1-3 PARLIAMENT SQUARE HERTFORD SG14 1EX
TELEPHONE (0992) 500919 FACSIMILE (0992) 554261

CIRCLE NO. 927

preset for a couple of dozen standard functions; `eg -esym(534, strcat, strcpy)`. I normally turn off all return mode checking with `-e534`; Perhaps I should review this practice.

The final difference in the comparison was in the wrap-up. It showed that an included header file was not actually needed. Lint V4 had an occasional blind spot on header inclusion. Although not causing any problems, loading headers obviously takes time and space, so removing unnecessary ones speeds up both linting and compiling.

After transferring my Lint V4 options to V5, I obtained almost the same behaviour from V5 as from V4. Almost. The `_cdecl` errors and extraneous header warning remained. After checking one module, I linted the complete set. The result was initially alarming, but I calmed down as I checked the messages, three of which were more unused include files. However, most problems came from a new warning: 644, 'Variable may not have been initialised'.

```
enum EFLG { BAD=-1, ZERO, GOOD, STOP };
typedef int BOOL;
#define TRUE  (BOOL) 1
#define FALSE (BOOL) 0

BOOL go=TRUE;
enum EFLG efl=ZERO;
int num;

  ....
  go=num;
  if(go) efl=5; else efl=1;

Lint comments:
  go=num;
tst.c(110) : Warning 632: Assignment to strong type (BOOL) in context:
             assignment

  if(go) efl=5; else efl=1;
tst.c(111) : Error 64: Type mismatch (assignment) (int/enum)
tst.c(111) : Error 64: Type mismatch (assignment) (int/enum)
```

Figure 5 - Strong Type Checking

```
/*lint -strong(AJX,FLAGS)*/
/*lint -strong(AiJX,FLAGS1)*/
/*lint -strong(AiJX,FLAGS2)*/

typedef unsigned FLAGS;
typedef FLAGS FLAGS1;
typedef FLAGS FLAGS2;

void xx(void)
{
FLAGS flg;
FLAGS1 flg1=128;
FLAGS2 flg2=256;

  flg=flg1;
  flg2=flg;
  flg1=flg2;
}

Lint comment:
  flg1 flg2;
f:x.c(17) : Warning 632: Assignment to
strong type (FLAGS1) in context:
    assignment
f:x.c(17) : Warning 633: Assignment from
a strong type (FLAGS2) in context:
    assignment
```

Figure 6 - Hierarchic Type Checking

Previous versions of Lint checked only that a variable was initialised on a line prior to where it was referenced. V5 goes further. If initialisation takes place only in a `switch` statement, for example, which has no default, Lint now issues a warning (Figure 2). The fact that no other case can possibly exist is unknown to Lint, so it forces you either to avoid the warning by adding a `default` or to turn the warning off. If the variable is uniquely named in the program, you could use `-esym(644, prvpag)`. Similar checks occur for `if`, `for` and `while`.

Another improvement in Lint V5 concerns promotion of variables in a function call. V4 accepted that automatic promotion of an `int` to a `long` occurs in a function call without a cast. V5 no longer accepts this. For example, if `int save_name(long)` is called with `save_name(0)`, the warning 'Significant prototype coercion (arg. no. 1) int to long' is issued. The same happens if the argument is, say, an `int`.

V5 also checks format incompatibilities on the `printf()` family of functions, reporting such inconsistencies as `printf("%+c", cc)` and numeric size discrepancies.

Lint V5 also has a more stringent approach to overflow of constants (Figure 3). You must now be very precise about how constants are defined and manipulated.

Lint V4 got upset over certain types of `switch case` and function terminations. A common example is using `exit()` to terminate a program. If this is in a `switch` or is the last line of a function which has valid returns elsewhere, V4 complained of a missing `break` or `return`. To satisfy it, a dummy break or return had to be added (Figure 4).

Lint V5 now knows about `exit()`, among others, and allows you to define similar functions of your own. This is an aspect of another new feature, *function mimicry*, whereby you indicate that a given function has similar attributes to another.

Strength

Every now and then I think 'Wouldn't it be nice if lint...'. One thing on my wish list was a tighter rein on what can and cannot be assigned to variables defined with `typedef` and `enum`. As long as the underlying type is the same, C enforces no further type checking; an `int` which is `typedef`'d as a `BOOL` is still, at heart, an `int`.

Figure 5 shows a (contrived) case where a variable is `typedef`'d to distinguish it from others. Normally, only the programmer's eagle eye ensures this. There is no Pascal-like variable typing in C. Until Lint V5. Add `-strong(AJXB, BOOL)` to Lint's options and there is a resounding warning. For `enum`'s the level is raised to that of error. Lint also supports strong typing of pointer-indexed variables.

Strong types have a hierarchy. Hierarchy typing allows variables of a given parent type to be manipulated in terms of this parent, but inhibits interaction between the child types. In Figure 6, `FLAGS1` can be combined with `FLAGS` but cannot be combined with `FLAGS2`.

A feature of Lint which is useful for multi-module projects is LOB. The idea is to make a Lint Object Module from each finished module and to use these when linting unfinished modules. This time-saving facility lends itself to incremental linting via a make utility.

There are many more Lint facilities than can be described here. For example, indentation checking, prototype generation, redefinition of scalar variable sizes, and so on.

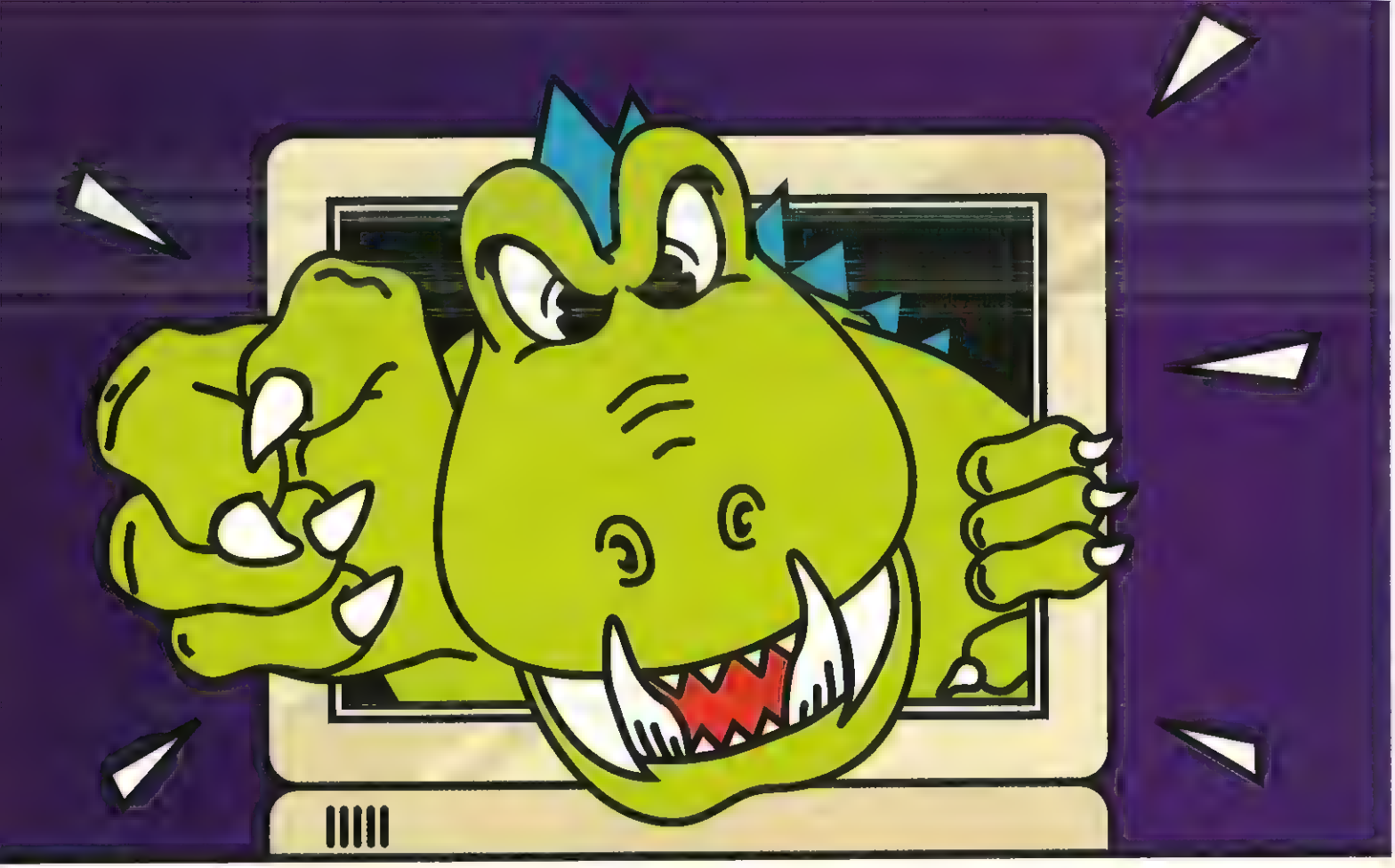
Conclusion

Gimpel's PC-Lint has always been an indispensable tool. Version 5.0 has not only addressed the odd slip-up in earlier versions, but has added an almost Pascal-like rigidity to type checking which I, for one, am grateful for.

EXE

Dave Stiles gave up a restful life of full employment to become a freelance programmer and writer. He now works twice as hard for half the money. His current hobby is sleeping until 10 am.

Gimpel's PC-Lint is available from tool suppliers such as Grey Matter (0364 53499) and Software Construction Company (0763 244114) at around £85 to £90.



Add more bite to your PC development

Operating systems, tools, languages, debuggers and compilers. Get your teeth into the latest products to help you develop better code faster at Software Development 92. You'll find all your requirements catered for in the new PC Development Forum.

And, while you're there, you can exchange ideas and learn techniques from the leaders in the technology including: Microsoft, Borland, Gupta, QA Training, Oracle and Novell. The people with answers to your questions.

PC DEVELOPMENT AND MORE

You'll not only see the largest array of PC development tools in the UK – the show is a multi-platform, cross-industry event that combines an exhibition with a multi-track conference programme that provides practical technical advice and relevant information that will make your software development faster and better.

ADDING MORE BITE PC DEVELOPMENT FORUM

Where you see this sign you'll find people to help you with PC development

SPONSORED BY

ProgramNOW

PC Development Forum supported by Microsoft and ProgramNOW

Dragon Critter supplied courtesy of Crittercal Computers Ltd

THE PERSONNEL TOUCH

The show is about people as well as products. There will be a People and Skills Centre provided by Computer People. And, if you're interested in getting together with people who think the way you do, there's a User Group Lounge where you can meet your peers and join a group that matches your needs.

SAVE £10

Send for your tickets now and we will waive the £10 admission fee. All you have to do is phone Lynne Davey on 081 742 2828 or fill in the coupon and return it to the address shown below.

ORDER YOUR TICKETS NOW
TICKET HOTLINE 081 742 2828

SOFTWARE DEVELOPMENT 92

The Software Tools Show

CIRCLE NO. 928

16-18 JUNE 1992, WEMBLEY CONFERENCE AND EXHIBITION CENTRE, LONDON

To save £10 return to: Lynne Davey, Software Development 92, Blenheim Online, 630 Chiswick High Road, London W4 5BG.

MR/MS _____ INITIALS _____

SURNAME _____

JOB TITLE _____

COMPANY _____

EXE/5/92

ADDRESS _____

_____ POSTCODE _____

TELEPHONE _____

☐ Please send me more information on the Software Development 92 Conference

BLENHEIM ONLINE

Blenheim Online, Blenheim House,
630 Chiswick High Road, London W4 5BG
Tel: 081 742 2828 Fax: 081 742 3182

Blenheim Online is a division of Blenheim Exhibitions & Conferences Limited

Terminally Asynchronous

Need to write Comms software?
Cliff Saran examines five alternatives to the BIOS and DOS services.

If change is a measure of progress then PC serial communications must be the computer industry's equivalent to the dinosaurs. When the PC came into existence in 1981, IBM could not possibly have conceived that it would evolve into the state-of-the-art, general purpose work-horse that it is today. We now have 486 machines which give comparable performance to the likes of minis. However, not all the silicon in the PC has been subjected to this technological bombardment - our RS232 ports are still driven by UARTs which offer more or less the same functionality as the modest 8250 that made its debut in the serial port of the very first IBM PC.

That's not all. The BIOS and DOS haven't improved their serial communications services significantly either. The BIOS provides routines to initialise the comms ports and the basic read/write character routines. It supports hardware handshaking, although the baud rate is limited to 9600. Characters are transmitted and received using polled I/O, which is inefficient since it suspends the main program until a character has been received. There are no facilities to provide Xon/Xoff software handshaking, nor are there any built-in file transfer protocols (eg Kermit or XModem). It isn't too difficult to write an alternative set of serial I/O routines that use interrupt-driven I/O (see *.EXE March 1992 - Efficient PC serial Communications*), although I wouldn't recommend that you try to re-create one of the file transfer protocols (macho programmers should look at *C Programmer's Guide to Serial Communications* by Joe Campbell for a painstaking description of the XModem protocol). This month I have been busy forging a first impression of five C comms libraries and here are my findings...

C Asynch Manager

Blaise's C Asynch Manager is a comms library that is based upon a three-level design. At the heart of the library there is Level 0, the lowest level of routines which talk directly with the UART. These routines can only be accessed from assembly language, using the 80x86's registers to pass parameters. This level contains routines which are similar to the BIOS comms services (ie initialise serial port, transmit/receive characters). However, unlike the BIOS, C Asynch Manager is interrupt-driven. Along with hardware handshaking, Level 0 also provides software flow control implemented using the Xon/Xoff protocol. Although Level 0 function can be linked with an application, Blaise also offers an alternative which installs Level 0 as a set of memory-resident routines which can be accessed using a software interrupt. The routines are permanently loaded into memory using a utility called LCOM.

```
lcom /i <intvector#> /q
```

The next layer is called Level 1 and this provides a C interface to Level 0. Characters are read from and written to separate input and output queues. The size of these queues can be specified when the port is opened. The Level 0 routines transfer data from these queues. Since a program executes far quicker than characters can be received by the UART, characters cannot be read from the input queue 'on the fly'.

	C Async Manager Library	Crystal Comm For Windows Library	Asynchronous Communications Library	Essential Communication Library	Comm-Drv
Manufacturer	Blaise 0101 415 5405441	Crystal Software 0101 906 8227994	Greenleaf 0101 214 4282561	South Mountain Software 0101 201 7626965	Willies' Computer Software Company 0101 713 4984832
Serial Card Support	PC/AT/PS2	PC/AT/PS2	PC/AT/PS2 intelligent boards from DigiBoard, Arnet & Star Gate	PC/AT/PS2 Intelligent boards from DigiBoard	PC/AT/PS2 DigiBoard COM/Xi
File Transfer Protocols	XModem, YModem	XModem, YModem, Kermit	XModem, YModem, ZModem, Kermit	XModem, YModem, ZModem, Kermit	Vendor specific
BackGround File Transfer	Y	N	N	Y	N
Max Baud Rate	19200	19200	115200	115200	115200
Hayes Support	Y	Y	Y	Y	N
Levels in Lib	3	1	3	2	1
Num of Lib calls	57	49	298	120	94
Mem Resident	Y	N	N	Y	Y
Price	£110 Grey Matter	$350	£175 Grey Matter	£150 Grey Matter	$89.95

Note that PC/AT/PS2 refers to support for the 8250/16450/16550 UARTs respectively. Prices quoted are for the source code versions of the libraries.

Figure 1 - Comparison of Comms Libs

THE PROFESSIONAL PROGRAMMERS' CHOICE

32-BIT

Zortech C++ for Windows, DOS, OS/2 & UNIX.

Now the world's first multiple platform C++ compiler is available from Symantec – Zortech C++ version 3.0 Release 4.

When it comes to building large, complex applications, C++ is the programmers' language of choice.

Zortech C++ for Windows, DOS & OS/2 is the only professional compiler to offer tools for Windows, DOS and OS/2 in an integrated package. Zortech C++ goes well beyond other compilers presenting not only professional tools but a collection of exclusive features specifically designed to help you increase your productivity and reduce your development time.

Zortech C++ offers a WINC library that enables you to easily convert DOS command line programs to true Windows applications. And you won't need to switch development systems, because you'll have the flexibility of moving across different platforms.

All of which means you'll get your applications to market faster than ever before.

"All in all Zortech has put together a winner at every level and offers a lot of complier for the money". **– Don Binns, Program Now, June 1991**

Developer's Edition **£399.00*** Science & Engineering Edition **£599.00***
UNIX Edition **£399.00*** Macintosh Edition **£249.00***

FACTS DIRECT

PHONE **0865 727 232**
FROM YOUR FAX MACHINE HANDSET – WHEN ASKED FOR A CODE ENTER:

◆

202 021 FOR **APPLICATIONS**

◆

202 022 FOR **UTILITIES**

◆

202 023 FOR **MAC UTILITIES** AND **GREATWORKS**

◆

202 024 FOR **THE SYMANTEC PRICE LIST**

For immediate information use your fax telephone to call Facts Direct on 0865 727232.

When asked for a six digit code enter (code number) or 202024 for your price list.

PROVEN GLOBAL OPTIMISER – Zortech C++ offers globally optimised compilers that boost program speed and significantly reduce program size. So regardless of the platform, Zortech C++ delivers performance and stability where it matters – in your finished user application.

DOS EXTENDERS – Zortech C++ even includes royalty free 32 bit and 16 bit DOS extenders, for the cost effective development of programmes with advanced memory requirements.

WINDOWS – Zortech C++ provides absolutely everything you need to edit, compile, link and debug any Windows application you won't need to buy Windows SDK. We've also added a help compiler and engine, a resource compiler, Windows header files and libraries, and printed SDK documentation.

TOOLS AND UTILITIES – with source code provided – help avoid re-inventing the wheel e.g. TSR's mouse drivers, memory tools. A list of 3rd party products which support Zortech C++ is included with the product, for example ZINC interface library and POET object-orientated database library.

FULL SOURCE CODE – to all standard libraries included.

Call NOW for more information

0628 777677

SYMANTEC.

Fax No: 0628 776775

I want the professionals choice for a professional result!

☐ **Please send me more information on Zortech C++ NOW!**

Name Position

Company

Address

Tel No.

.EXE 5/92

Symantec (UK) Ltd, MKA House, 36 King Street, Maidenhead, Berkshire, SL6 1EF

CIRCLE NO. 929

PVCS

Version 5.0

Comprehensive Software Configuration Management

Due 1st May '92

	DOS,OS/2,UNIX UPGRADE		DUAL(DOS & OS/2) UPGRADE	
Configuration Manager	£95	£299	£125	£399
Configuration Builder	£45	£135	£59	£175
PVCS Editor	--	£159	--	£199
Gateway	--	£109	--	--

To order or for more infomation call :

Readmar Systems

071 625 5255

Visa/Access

Authorised VAR

CIRCLE NO. 967

Microprocessor Development Tools

EMULATORS, SIMULATORS, COMPILERS, ASSEMBLERS, PROGRAMMERS

77C82, 8085, Z8, 68000, 6051, 32010, 68HC11, 6301, 6502, 6805, Z80, 6809, 8096, 740 Series, 7720, MIPS R2000, etc . . .

SMAC Universal Assembler

NEW

- ✔ Relocatable
- ✔ Fast Assembly
- ✔ Caters for all Microprocessors and Controllers with bus widths from 1 to 255 bits wide
- ✔ Instruction sets for many microprocessors included - from Z80 to RISC 2000
- ✔ Instruction set compiler *AND VERIFIER* included - add your own micro
- ✔ Iterative macros
- ✔ Intelligent jump facilities
- ✔ Linker and *MAKE* facility

ROM Emulator

- ✔ Covers ROM sizes from 16 Kbit to 8 Mbit
- ✔ Full screen editor
- ✔ Emulates all 24, 28, 32 & 40 pin devices
- ✔ Fast download - loads 1 Meg. in under 5 seconds
- ✔ Split & Shuffle capability

2 Field End • Arkley • Barnet • Herts EN5 3EZ

Telephone : 081-441 3890

SMART COMMUNICATIONS

CIRCLE NO. 968

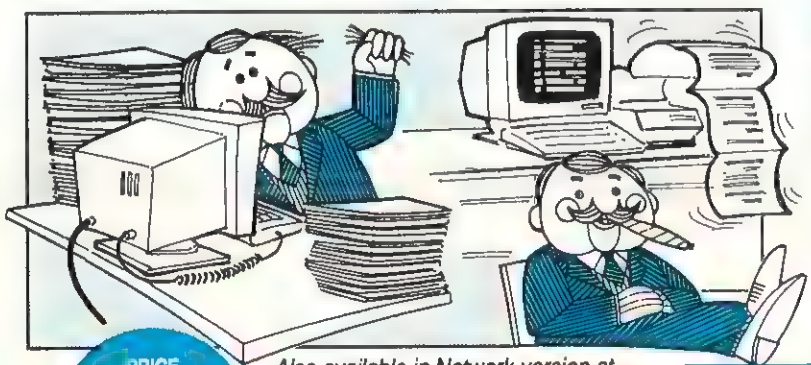
Put your PC on Autopilot . . .

AUTO RUN

PC Automation Software

Automate your PC to carry out all your repetitive tasks. For example, set Autorun to run Month End routines, Print reports, Sort databases, perform Mailmerges, Import and Export Data or Files across networks, Log onto Mainframes via modems, backing-up or processing data automatically at any time, day or night.
Autorun automation software will start-up, perform all assigned steps and terminate, no matter how complex, totally unattended.

- Totally unattended operation
- Interactively learns the keys you press and the time between each keystroke
- Remembers Control, Alt and Shift keys
- Decision making capabilities within tasks
- Executes your tasks at any time, day or night
- Power failure recovery during automated processes
- Automatic Day and Date function
- Maintains time delay between keys
- Unlimited number of jobs and keys
- Fast and convenient editing of keystrokes



PRICE £99.95 + VAT

Also available in Network version at

£299.95 Excl.

TURN YOUR PC INTO YOUR PERSONAL ROBOT!

WHAT THE PRESS SAY . . .

"Autorun is extremely easy for first time users to operate, the added features available to more experienced operators make it a very powerful suite of programmes indeed" — ***Micro Computer Mart,*** *April 1991*

"A lot of man hours can be freed up by this program . . . Autorun is a worthwhile program that will find a welcome home in many offices."
— ***Computer Shopper,*** *October 1990*

"Autorun adds up to a very powerful (and rather clever) time saving utility."
— ***What Micro,*** *April 1991*

UK DISTRIBUTORS:-

GUILDSOFT LTD.

GUILDSOFT DEPT.EXE 2THE COMPUTER COMPLEX, CITY BUSINESS PARK, STOKE, PLYMOUTH, DEVON PL3 4BB

PHONE 0752-606200

GUILD Soft

CIRCLE NO. 930

When you need to read a character from a comms port it is necessary to synchronise the execution of the program with the rate at which characters are received. Level 1 provides `iwait_a1()` which halts program execution until a character has been put in the input buffer. Conversely, it is sometimes necessary to synchronise the transmission of characters. Blaise provides the `drain_a1()` routine for this purpose. `drain_a1()` waits until all characters have been transmitted before returning to the program.

The topmost layer in C Asynch Manager is Level 2. This contains a set of routines that are supposedly easier to use than their Level 1 counterparts, although this is questionable. Level 2 doesn't offer any substantial improvements over and above Level 1 so why should you use it? The only improvement worth a mention is Level 2's straightforward approach to port configuration. In Level 2 Blaise has thoughtfully provided a structure which enables all options to be changed in one go, while in Level 1 it is only possible to change one setting at a time.

C Asynch Manager supports both the XModem and the YModem file transfer protocols. It uses a complicated structure (`FTCS`) to hold all the information required to perform a file transfer. Since `FTCS` contains all the info needed to perform a file transfer, Blaise has designed the Upload/Download routines such that they can run in the background. This means that an application does not have to suspend execution until a file transfer has completed.

C Asynch Manager is able to control a Hayes compatible modem. It provides several routines including functions for picking up the receiver, dialling, auto-answering and hanging-up. Commands can be sent to the modem as strings using the `cmd_hm()` routine.

CrystalComm

CrystalComm for Windows from Crystal software is the only library of the five that is distributed as a DLL and can be used to develop Windows-based comms applications. There is also a version of the library available for DOS. Unlike C Asynch Manager, CrystalComm is organised as a single layer of routines. Almost all routines in the library take a comms port handle as their first parameter. The one exception is `ComOpen()` which returns such a handle.

Serial communications is interrupt-driven so it is necessary to check whether there are characters in the input queue before reading it. CrystalComm doesn't provide a routine which suspends program execution until a character has been received. Instead it contains a function (`ComStatus()`) which returns the number of characters in the input queue. In order to synchronise program execution with serial input it is necessary to repeatedly call `ComStatus()` until it returns a positive integer.

While polling, the program can perform other chores - characters will not be lost unless the input queue overflows. As an alternative, CrystalComm also provides `ComReadTimed()` which waits for a character to arrive until a pre-defined timeout occurs. The review copy of this library doesn't allow the programmer to check whether the output queue is empty. However, a new version of CrystalComm will be available shortly which will provide this facility.

There are three methods of file transfer available to the programmer. Unlike C Asynch Manager, CrystalComm supports the Kermit file transfer protocol. The function `ComSetKermitParams()` allows you to tweak the various options available. However the default values seem to work pretty well. CrystalComm also supports the XModem and YModem file transfer protocols although.

CrystalComm also provides routines for controlling a Hayes compatible Modem. Unlike C Asynch Manager, CrystalComm doesn't allow the programmer to specify a Hayes command string directly.

Glossary

Asynchronous Communications - A type of serial communications in which each character is passed from the transmitter to the receiver as an independent entity, consisting of a *Start Bit*, up to two *Stop Bits* and an optional *Parity Bit*.

Baud Rate - The frequency at which electrical impulses are sent down the communications line.

Bits Per Second - The number of data bits transmitted per second.

Checksum - A method used to determine whether a block of characters has been received correctly. This normally involves summing the ASCII values of all characters received and comparing the least-significant byte of the result with a separately transmitted checksum byte.

CRC - (Cyclic Redundancy Check) A more powerful method for detect transmission errors than *Checksum* (for a detailed explanation see *C Programmer's Guide to Serial Communications* by Joe Campbell).

CTS/RTS - These two signals provide a *Handshaking* mechanism for serial communications. The transmitter sends Request To Send signal to the receiver which responds with a Clear To Send if it is ready to receive.

Download - This is the process of receiving one or more files from a host computer (see also *Upload*).

Full-Duplex - The ability to transmit and receive information simultaneously (see also *Half-Duplex*).

Half-Duplex - Communications can occur in both directions but it is limited to one direction at any given moment in time (see *Full-Duplex*).

Handshaking - A mechanism which enables the receiver to remain in synchronisation with the transmitter (see *CTS/RTS*).

Hayes Compatible - Modem conforms to the Hayes communication standard and can be controlled using a set of instructions known as the AT Command Set.

Interrupt Driven Comms - This provides the most efficient way of transmitting and receiving serial data. It effectively enables serial communications to occur in the background while the main program runs in the foreground.

Kermit - This is a file-transfer protocol which enables batches of files to be transferred to or from the host (see *XModem*).

Parity - Error-detection that works on a per-character basis. A Parity bit is an addition bit that is appended to the end of a character to be transmitted in order to make the resulting byte hold either an even or odd number of bits set to one.

Polling - A simple method of achieving serial communication which relies on the software continually checking the *UART's* status register to determine whether it is ready to transmit or receive a character (see Interrupt-driven Comms).

RS-232 - The Electronics Industries Association (EIA) standard which describes the behaviour of the electrical signals needed to perform serial I/O.

Start Bit - This is a synchronisation pulse which informs the receiver that the first bit of a character will be transmitted next (see *Stop Bit*).

Stop Bit - Another synchronisation pulse which tells the receiver that the last bit of a character has been transmitted. There may be 1 or 2 Stop Bits (see *Start Bit*).

UART - (Universal Asynchronous Communications Adapter) A device which controls serial communications.

Upload - This is the process of transmitting one or more files from a host computer (see *Download*).

XModem - A file transfer protocol which enables a single file to be *Uploaded/Downloaded*. Data is transferred in packets of 128 bytes and each packet is associated with *Checksum* or *CRC*.

Xon/Xoff - A protocol which lets the receiver control the flow of characters down the transmission line. If the receive buffer becomes full, the receiver sends a special Xoff character to inform the transmitter to stop sending characters. When it is able to receive again, the receiver sends an Xon character.

YModem - This is an enhancement to the *XModem* protocol which enables multiple files to be transferred. The packet size is increased to 1024 bytes and there is an extra packet at the start of the transfer which consists of a file list and size and date information.

ZModem - This is a further extension to the *XModem* family of file-transfer protocols and provides rapid file transfer.

CommLib

Greenleaf's CommLib is based on a three level hierarchy. However, unlike Blaise, Greenleaf takes full advantage of the three layers. The lowest level, Level 0, talks directly to the hardware. Level 1 provides a C interface to Level 0 and Level 2 enables the programmer to write device-independent software. Unlike the two previous libraries, Commlib contains several device drivers in addition to the 8250 driver. These include the Arnet, DigiCHANNEL and Star Gate intelligent serial cards together with a number of non-intelligent cards. The Level 2 library routines provide a consistent, device-independent interface to these device drivers.

CommLib maintains this device-independent interface by providing a `PORT` data structure that holds pointers to the routines which apply to a given serial card. When a function such as `ReadChar()` (read a character from serial port) is called, CommLib uses the `PORT` structure to invoke the correct device-specific routine which is then used to perform the actual read operation.

The input buffer must be polled in a manner similar to the method adopted in CrystalComm to check when a character has arrived. However Greenleaf also allows the programmer to determine whether the output buffer is empty. In addition, it provides routines which return how much space is free in the receive and transmit buffers.

CommLib offers the programmer an impressive list of built-in file transfer protocols. In addition to providing several variants of the XModem/YModem and Kermit protocols, it also enables files to be transferred using ZModem. Instead of providing a structure to hold the settings of the various options for each file transfer protocol, CommLib uses different versions of the library calls. For instance, the function `XmodemlKCRCSend()` will transmit a file using XModem with a 1 KB packet size and CRC error checking. On the other hand, `XmodemChecksumReceive()` will transmit the same file using XModem with a 128 byte packet size and checksum error checking. There are four versions of XModem, three versions of YModem, two versions of Kermit and two versions of ZModem.

In addition to supporting the usual modem controls (eg Dial, Hang-up), CommLib provides a comprehensive list of routines for controlling the modem. CommLib also includes a number of routines for writing terminal emulation or Bulletin Board Service (BBS) software. There are functions for controlling the keyboard and a set of functions for providing a 'no frills' DOS-based windowing system.

Essential Comms

The Essential Communications Library (ECL) from South Mountain Software is a two-level library that supports a number of serial port cards. ECL's equivalent to Blaise's Level 0 routines is the XCommS (eXecutive Communication Server) - a rather extravagant name for a set of routines which talk directly to the hardware (ie device drivers). However, unlike C Asynch manager, it is not possible to access these routines from C. XCommS may be linked with an application or, alternatively, it can be loaded as a memory-resident module using the XCOMMS utility. The command-line below will cause XCommS to load into memory, set aside

THE SOFTWARE CONSTRUCTION COMPANY

BORLAND

Borland C++ v.3.0	£189
C++ & Paradox Engine	£199
C++/App. Framework	£289
Turbo C++ v 3.0 New	£45
Turbo C++ for Windows	£79
Turbo Debugger & Tools	£85
Turbo Pascal v.6.0	£65
Pascal Professional	£129
Pascal for Windows	£99
ObjectVision v.2.0	£79
Paradox 3.5	£325
Paradox Engine	£189
dBASE IV 1.5	£325
dBASE IV 1.5 Upgrade	£90
World of C++ Video	£50
Word of C++/Turbo C++	£80

UK AUTHORISED

COMMUNICATIONS LIBRARIES

AdComm	£190
Async Professional	£90
ASYNCH PLUS	£125
C ASYNCH MANAGER	£125
C Comm Tlkt Prof Ext DOS	£325
C Comms Toolkit Prof.	£195
C Communications Toolkit	£99
Essential Communications	£215
GET-IT	£99
Greenleaf Comm++	£129
ProBas Telecom Toolkit	£49
QuickComm	£99
SilverClip SPCS	£225
SilverWare C Asynch Lib.	£165
Solid Link with source	£325

DATA FILE HANDLERS

B+Tree Data Manager	£65
B-Tree Filer	£90
B-Tree Filer (multiuser)	£125
BTree/ISAM multi user	£115
BTree/ISAM single user	£75
Btrieve for DOS	£385
C++ Database Class Lib	£150
C-Data Manager w/source	£385
C-Index/II	£449
C-ISAM	£215
c-tree Plus	£385
c-tree with r-tree	£419
CBTREE	£129
CodeBase 4.5	£255
CodeBase 4.5 Upgrade	£95
db/LIB Database Prof Lib	£125
BFtrieve	£115
dp-MAX	£99
E-Tree Plus	£165
Essential B-Tree	£129
EZedit Unlimited Users	£195
Faircom Server	£190
Phase One	£319
Pinnacle Engine	£165
POET	£259
QBE/Phase 1	£255
Toolbox Professional	£835
Toolbox Special Edition	£515
Topaz - Multi-user DBMS	£99
r-tree Report Generator	£190

GENERAL FUNCTIONS

"C"erious Toolkit Plus	£65
3PX Extended Function Lib	£129
BASIC Development Tools	£65
Blackstar C Function Lib	£65
C TOOLS PLUS	£99
C Utility Library	£165
dBUtility Library	£100
FUNCky.LIB	£175
Grumpfish Library	£129
Object Professional	£125
ProBas Prog. Library	£115
ProBas Toolkit	£85
QuickPak Professional	£129
Rogue Wave C++ Class Lib.	£59
Soft.Clip (Net. version)	£195
Spindrift Library	£99
Spontaneous Assembly	£129
Super.LIB Prof. Network	£155
Super.LIB Professional	£129
SuperFunctions	£195
Tools.h++	£129
Turbo Professional	£89

Microsoft

BASIC PDS 7.1	£215
Visual Basic,	£90
C/C++ 7.0 Compiler	£215
C/C++ 7.0 Upgrade	£99
C 6.0 & Windows SDK	£315
COBOL 4.50	£420
FORTRAN 5.1	£199
MASM 6.0	£70
MS-DOS 5 Upgrade	£49
Mulimedia Dev. Kit	£215
Pascal Compiler	£135
QuickBASIC	£49
QuickC	£49
QuickC/Assembler	£75
QuickC for Windows	£90
QuickPascal	£49
Source Profiler	£39
Test for Windows	£179
Visual Basic Toolkit	£129
Windows 3.0	£65
Windows S.D.K.	£215

U.K. Authorised

GRAPHICS LIBRARIES

3-D Computerscape/source	£189
3-D MiniCAD w/source	£450
ChartBuilder	£99
dGE	£195
EGA Toolkit	£159
Essential Graphics Chart	£905
Essential Graphics Kernel	£129
Flipper	£190
FontWINDOW	£99
Geograf Prof. Ed.	£210
GFORCE	£165
GFX Font & Icon Editor	£65
GFX Fonts & Menus	£99
GFX Fonts/Menus/Graphics	£165
GFX Graphics	£99
Grafmatic	£165
GrafPrint	£129
Graph Link	£129
GraphiC	£300
Graphix	£169
GraphPak Professional	£99
GSS Graphics Dev. Tlkt.	£515
GSS*CGM Metafile Interp.	£319
GSS*GKS Kernel System	£515
GX Developers Pak	£435
GX Effects	£129
GX Effects w/Source	£259
GX Graphics	£129
GX Graphics w/source	£259
GX Text	£99
GX Text w/Source	£195
Halo F/X	£195
Halo Image File Format	£165
HALO Professional	£385
INGRAF	£229
MEGAVDI	£579
MetaWINDOW	£190
MetaWINDOW/Plus	£255
MetaWINDOW/Premium	£385
NDP Plot	£200
PC Graphics Library	£385
PC-VDI	£255
PCX Prog Tlkt w/source	£389
PCX Programmer's Toolkit	£165
PHONTM-The Fontmaker	£255
ProGraphx Toolbox source	£259
Quick Geometry Library	£129
QuickWINDOW/C	£99
Turbo Halo	£65
TurboWINDOW/C	£99
TurboWINDOW/Pascal	£99
VID and DIG Graphics Lib	£259

GRAPHICS WINDOWS

Essential Graphics GUI	£195
Font-Tools w/source	£165
graphics-MENU	£165
graphics-MENU Pro Pak	£325
GSS XVT for DOS	£385
Object-Menu	£239
Object-Menu with source	£579
T-Windows with source	£190
TEGL Windows Toolkit II	£65
Zinc Interface Library	£129

COMMUNICATIONS & LANS

dB/LIB Network	£375
Net Lib	£195
Netware C Interface	£190
Network System Calls Clip	£190
Network Toolbox/N	£90

SHAREWARE FOR DEVELOPERS

ASM Library (20)	£40
C/C++ Library (68)	£66
dBASE/Clipper (87)	£100
DOS Utilities (37)	£40
Novell Netware (54)	£66
Turbo Pascal (38)	£40
Visual BASIC (11)	£40
Windows (64)	£100

(Number of 5.25 disks)

TEXT SCREENS

C-Worthy Library Source	£175
C-Worthy Upgrade to v.2.0	£125
C-Worthy/cwArchitect	£389
C-Worthy/cwArchitect/Srce	£559
Code Screen	£65
cwArchitect Source Code	£515
DataWindows	£255
HIM	£225
MEWEL Window System	£190
PANEL Plus II	£319
PANEL Plus II Util Source	£775
ProGraphx Menus source	£129
QuickWindows Advanced	£99
Vermont Views + Designer	£319
Vermont Views GraphEx	£99
Vermont Views QV	£165
Vermont Views with source	£575

386 COMPILERS & EXTENDERS

386 C Code Builder	£449
386|ASM/LinkLoc	£635
386|DEBUG	£129
386|DOS-Extender Dev. Kit	£319
386|VMM	£190
C Network Compiler/386	£645
C++ Developer's Edition	£399
EDU-16	£319
F77L-EM/32 w/OS/386	£899
Fortran 77 386/486	£645
LinkLoc	£515
Metaware High C 386/486	£645
MetaWare Pascal 386/486	£579
NDP C++	£545
NDP C-386	£545
NDP C-486	£750
NDP C-SX	£350
NDP Fortran-386	£545
NDP Fortran-486	£750
NDP Fortran-SX	£350
NDP Pascal-386	£545
OS/286 Developer's Kit	£449
OS/386 Developer's Kit	£449
Smalltalk/V 286	£129
SVS C'ANSI C	£775
SVS Fortran-77/386	£965
SVS Pascal/386	£775
WATCOM C 8.5/386 Prof	£515
WATCOM Fortran 77/386	£515

OTHER C COMPILERS

C-terp	£195
Instant C	£319
Metaware High C DOS x86	£319
TopSpeed C - Professional	£169
WATCOM C 8.5 Professional	£319

DATABASE MANAGEMENT

Alpha Four	£345
Clarion Personal Dev. 2.0	£79
Clarion Professional Dev.	£679
PALCOM Network	£645
PALCOM Single User	£319

xBASE

Clipper 5.0	£325
FoxBASE+	£295
FoxPro	£495
FoxPro/LAN	£695
Quicksilver	£389

OBJECT-ORIENTATED

Smalltalk/V	£85
Smalltalk/V 286	£129
Smalltalk/V Windows	£325

PASCAL

MetaWare Prof. Pascal x86	£385
Stony Brook Pascal+	£190
TopSpeed Pascal - Prof.	£109

UPGRADE TO MICROSOFT C/C++7.0

Microsoft C (any)	£99
Borland/Turbo C++ (any)*	£99
Zortech C++ (any)*	£99
Topspeed C++ (any)*	£99

*REQUIRES PROOF OF PURCHASE BEFORE 1ST FEBRUARY 1992 . ALL PRICES + VAT

*All prices exclude V.A.T.
REMEMBER WE SHIP FREE TO U.K. VISA or ACCESS accepted. Credit to approved accounts. New **Toolbox 5.0** catalogue now available, call. Please check prices when ordering and specify disk size. 3.5" shipped as standard.

1 THE MALTINGS, GREEN DRIFT, ROYSTON, HERTS. SG8 5DB
TELEPHONE (0763) 244114 FACSIMILE (0763) 244025

eight receive buffers and use COM1 with Xon/Xoff flow control.

```
xcomms 8  com2x
```

Access to Level 0 is achieved through the Level 1 C interface code. Level 1 provides drivers for five serial port cards which include the DigiBoard PC/Xe intelligent board and a number of non-intelligent boards with multiple serial ports. It also contains a generic driver which lets you configure ECL to work with other, unsupported multiple serial port cards. For instance, the following code extract will provide support for two comm ports on IRQ3, with COM2 at address 0x2F8 and COM3 at 0x3220. This is the configuration of the PS/2's dual async adapter.

```
xc_entr(5);
setport(0x2F8, 2, 3, COM2, 0);
xc_cport(COM3, 0x3220);
```

Initialising a comms port is perhaps a little peculiar - you have to tell ECL to operate with interrupt-driven I/O, otherwise it assumes that you intend to use the BIOS serial communications service. You also have to indicate how many input buffers your application requires. Unlike other libraries in this review, the size of the input buffer is fixed at 1 KB.

South Mountain Software has taken Blaise's approach to synchronising serial communications. It provides routines which suspend program execution until a character has been successfully transmitted or received.

ECL supports the XModem, YModem, ZModem and Kermit file transfer protocols. The various options are passed as parameters to the appropriate routine. This is more elegant than providing several versions of each routine (as in CommLib), although it is less flexible than Blaise's `FTCS` structure.

There is Hayes-compatible modem control and support for basic keyboard and video I/O. A unique feature of ECL is that it supports the emulation of three types of terminals (VT52, VT100 and ANSI). The `terminal emulation routines` use a `term_info` structure to describe the physical characteristics and behaviour of a given terminal.

Comm-Drv

Comm-Drv is a basic communications library from Willies' Computer Software Company (WCSC). There are three versions of the library. The first is a replacement for the BIOS comms services. It uses a memory-resident module called CommTSR to provide the serial I/O services. The module can be accessed from assembly language by invoking interrupt 14. For instance, the code below will send an asterisks character to COM1.

```
mov AH, 01h
mov AL, '*'
mov DX, COM1
int 14h
```

Next there's an MS-DOS device driver. This enables a programmer to access Comm-Drv using standard Open, Close, Read, Write and IOCTL calls. The last version of Comm-Drv is a C interface which contains an embedded form of the CommTSR.

Comm-Drv uses the port number to access the serial port. When it is time to configure the port, Comm-Drv adopts a similar mechanism to C Asynch Manager. It provides a `port_param` data structure and the `ser_rs232_setup()` routine for configuring the port using the values contained within this structure. There is also a routine for reading the current configuration of the port.

How much longer can you afford to wait?

Create Overlaid Programs-Fast.

BLINKER™, the world's first and fastest dynamic overlay linker, reduces your link time to seconds and reduces program memory requirements. Now you can use one linker for all your software projects.

One Linker, Many Languages.

BLINKER 2.0 links and automatically overlays DOS programs written in Microsoft® C, BASIC, Assembler, QuickBASIC™, Fortran, Pascal, Watcom™ C, Zortech C++, Clipper®, FORCE® and in Borland® C, C++, Assembler, and more.

Save Time and Memory.

BLINKER removes the need for overlay structures, simplifies program design and reduces memory requirements to save you time, effort and memory.

Memory Swap Function.

BLINKER is the ONLY linker to offer an integrated memory swap function, so you can run other large programs from within your program, with negligible memory overhead.

BLINKER™
High Performance Dynamic Overlay Linker

Don't Settle for Less.

Other major features include full CodeView® support, use of EMS/XMS at program run-time, and enhanced execution speed of overlaid code.

Time is Money.

BLINKER offers all this in a fraction of the time it takes to link with your current overlay linker. You know time is money, and link time is no exception.

Free Demo

To try our free demo on your own code and for more information contact our U.K. distributor:

Call: +44-81-994-4842 or
FAX: +44-81-994-3441

QBS Software Ltd.
10, Barley Mow Passage
London W4 4PH

BLINKER is available in 5.25" or 3.5" diskette format

Price £199
plus shipping & handling

VISA MasterCard

Blinkinc
P.O. Box 7154
Richmond VA
23221

CIRCLE NO. 934

Comm-Drv provides its own file transfer mechanism. The usefulness of this protocol is questionable since it is necessary for the remote machine to be running `ser2xfer_slave()` in order to transfer files. WCSC does offer a support library for Comm-Drv called Comm-Log which provides the programmer with background XModem and YModem file transfer.

This can be purchased separately for $89.95. WCSC also offers Comm-LOG and Comm-Drv together for $159. There is no modem support.

Conclusion

There's no need to list the similarities between these five libraries, as I'm sure you have already browsed thorough the table in Figure 1. There are a few notable differences worth a mention.

The XModem family of protocols was introduced in 1977 by Ward Christensen and it is historically significant because it was one of the first to be used for transferring files between IBM PCs over phone lines. Before XModem, Kermit was the most widely used protocol. Even today, there are some systems that only support Kermit file transfer.

If you want your comms software to operate with most computer systems, Kermit support is essential. This prevents you from using either C Asynch Manager or Comm-Drv.

ZModem was developed in 1986 by Telenet and offers significant enhancements over XModem and YModem including compatibility with satellite links and packet switched networks. Writing software for an uncertain future is a daunting task. However, if you're planning on supporting ZModem file transfer, it is likely that your communications software should survive for at least another decade. There are only two libraries in this review that support ZModem and these are Greenleaf's CommLib and South Mountain Software's Essential Comms Library.

Perhaps you're simply looking for better serial I/O. In this arena, the five libraries clearly offer a far superior interface to the BIOS. The question here is, which one is the easiest to use. The first stumbling block is in the quality of the documentation. C Asynch manager, CommLib and ECL come with extensive, well presented reference manuals. My first impression of the CrystalCOMM reference manual is that it was an afterthought - it provides only the barest of information needed to use the library. Comm-Drv is only slightly better.

Next there's the design of the library. The more levels of library call you have, the easier the library is to use. Finally there's support for non-standard hardware. A library that provides built-in support for specific hardware is certainly better than one that provides no support at all.

My personal favourite is Greenleaf's CommLib, with ECL a close second. As a final thought, whichever library you decide is best for you, I can guarantee that it will offer significant improvements over the ancient BIOS serial I/O services.

EXE

Grey Matter can be reached on 0364 53499

MKS RCS When a change . . .

INCLUDES FREE MKS MAKE

. . . isn't as good as the rest

Nobody gets it right first time.

Sometimes you do, but then throw it all away by making so many of those "subtle" little changes, you end up stuck with something that looks nothing at all like how you wanted it to.

Still, that's life. Or at least that's life before MKS RCS - the Revision Control System that keeps a complete history of all the changes made to your files. It lets you retrieve any, or all, of them without any fuss. You can retrieve them by date, by release number, or by whatever name you had previously given them.

Binary or text files? They're handled by MKS RCS with the same contemptuous ease.

Multi-users? No problem. A locking feature prevents more than one person from ringing the changes at any given time.

Branching? You can have as many parallel branches as you like, meaning you really will be able to see the wood for the trees.

Other features? File-compression; a new menu interface; conversion facility from SCCS files; compatible with UNIX system RCS.

Change away to your heart's content. If you *did* get it right first time (or the second, or the third, or the fourth...), you can go back to it at any time you like.

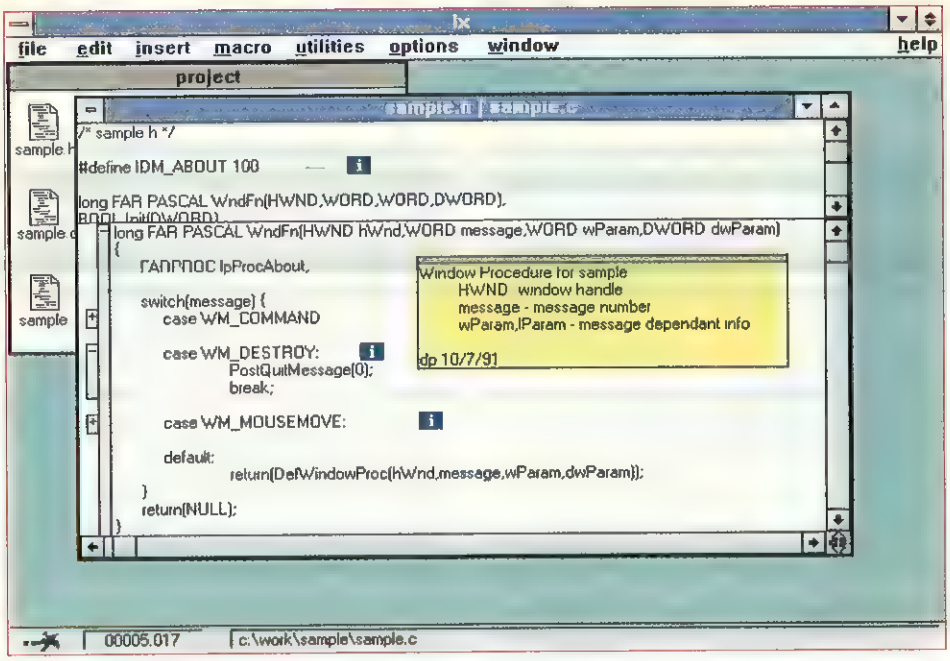
But there's one change you'll want to make permanent. And that's the change to MKS RCS. Call 0763 244144 for details.

Compatibility: MKS RCS is available for DOS, OS/2, XENIX and 386/UNIX.
For an MKS brochure contact the sole UK distributor:
The Software Construction CompanyLtd, 1 The Maltings, Green Drift, Royston, Hertfordshire SG8 5DB. Telephone: 0763 244114. Fax 0763 244025
MKS RCS, leader in performance AND price: DOS version £165 (5-user licence £645); OS/2 version £225; DOS & OS/2 version £259.

CIRCLE NO. 936

ix

programming editor for Windows



Full Windows implementation

Cut & Paste, nearly infinite Undo, Point and Click or key strokes control all features

Multiple files and windows using MDI

Load and view several files simultaneously in separate windows

Folding text

Manage large object modules with sections of text 'folded' up into a single line

Code documentation

Add 'post-it' style notes to any file

Fully reconfigurable keyboard and menus

Re-assign all keys and menus, make multiple key assignments

Powerful macro language built-in

Create and compile macros using any editor function with a Basic-like language

£95 + VAT

As more and more users switch to Windows 3, the demand for Windows software continues to grow. Yet much of the developer's toolkit still runs under DOS. Writing for Windows has always been more complex than for DOS, and project management can be especially tricky as Windows programs tend to be made up of large numbers of files.

ix was developed to enhance programmer productivity. ix runs under Windows, and uses all the standard GUI features – MDI, split windows, cut & paste, choice of fonts and colours. It also contains several unique features which allow the developer far greater control.

With ix you can edit and test within Windows – no more switching in and out of a DOS editor. Compilation/make files can be launched from a 'Browse' window.

Browse windows help you keep track of the multiple files of a Windows project. A browser can be created for each project and used to fire up edit windows, launch resource editors (such as SDKpaint) and run programs as in Program Manager.

'Post-it' style notes can be stuck onto any source file – without modifying the file. Their names can be used to give a hypertext-like help system throughout your project.

And there's more; our AdHoc macro language, reassignable keys, custom menus and free telephone support, but we haven't got space to tell you about all that here. For more information, call or write us and we'll send you our information pack.

• positive limited •
• 22 westminster buildings •
• new york street •
• leeds ls2 7dt •
• 0532 343 104 •

positive

Let me list the ways

Michael J Marshall explains how the humble linked list has saved him many hours of development time.

One of the most important rules of programming is to design for reusability whenever possible. This means: avoid writing purpose-built structures when a more generic structure already exists. Existing code has already been tested (probably in many different applications), so placing it in new code effectively removes a possible cause of bugs when testing the code. And since `TIME == MONEY` (according to my boss), the less time spent coding and debugging, the better.

One of the most useful data structures I've come across is the linked list. Every single programming book I've encountered has at least one implementation of linked lists. Unfortunately, a lot of these books underplay the usefulness of this simple container. It can be used to hold any number of effectively arbitrary objects and allows simple sequential retrieval of those objects. Implementation of linked lists is also quite trivial in most computer languages, which is a major plus - the simpler the code, the less likely it is to have hidden bugs.

Linked lists will benefit any program that needs a dynamically sized container with sequential access. If speed and size are an issue, you can replace the generic list with a purpose-built optimised version after you've got the rest of the application debugged.

```
/*
** SLIST.H
** Linked List Header File
** Michael J Marshall - X-ON Software
*/
#if !defined( _SLIST_H )
#define _SLIST_H
#if !defined(OK)
#define OK  0
#define FAIL  (-1)
#endif
/* Macros */
/* Returns no of elements in the list. */
#define slCount(list) \
   ((list)?(list)->count:0)
/* Short forms for traversing a list. */
#define slFirst(list) slIterate((list),1)
#define slNext(list)  slIterate((list),0)
/* Type Definitions */
typedef void (*free_t)( void * );
typedef int (*compare_t)( const void *,
                          const void * );
typedef struct SLNode {
  struct SLNode *next;
  void *datum;
} SLNode;
typedef struct SList {
  SLNode *head;
  SLNode *tail;
  SLNode *curr;
  unsigned count;
} SList;
 * Prototypes */
SList *slNew( void );
void slFree( SList *list, free_t del );
void slClear( SList *list, free_t del );
int slInsert( SList *list, void *datum );
int slAppend( SList *list, void *datum );
void *slGet( SList *list );
void *slIterate( SList *list, int start );
void *slSearch( SList *list, void *key,
                             compare_t cmp );
int slDelete( SList *list, void *data );
void *slSelect( SList *list,
                unsigned pos);
void *slCurr( SList *list );
#endif
```

Figure 1 - SLIST.H header file

```
/*
** SLTEST.C
** Linked List Test Program
*
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "slist.h"
void main( void )
{
  SList *list;
  char *str;
  list = slNew();
  if ( !list )
  {
    puts( "slNew() failed!" );
    return;
  }
  for ( ;; )
  {
    char buf[80];
    printf("Enter a string, \
<ENTER> to stop: " );
    gets( buf );
    if ( !buf[0] )
      break;
    /* NB Append a copy of the string */
    slAppend( list, strdup( buf ) );
  }
  printf( "\nThere are %d strings in \
the list:\n", slCount( list ) );
  for ( str = slFirst( list );
        str; str = slNext( list ) )
    puts( str );
  /* Free the list and all of the
     strings in it */
  slFree( list, free );
}
```

Figure 2 - SLTEST.C source file

The SList Package

SList is my implementation of linked lists. I wrote it a couple of years ago, and although I have added a few new features to it and improved its performance a bit, it is stable and reliable. It is easily the most useful element in my library.

The package itself is quite simple. It stores pointers to data created by the application, and returns them on request. You can add data at the beginning or end of the list, allowing simulation of queues and stacks. You can iterate over the list's elements, search the list for a specific datum, and remove items from the list.

Implementation

The interface to the SList package is defined by the macros and functions provided. The implementation is irrelevant to the user as long as the semantics of the defined interface are maintained. This encapsulation definitely helps maintenance, because if I completely rewrite the implementation of SList, the most work I will have to do on other code that uses SList is recompile it (because of the macros) - not a single line of code should need to be changed.

For return values, I follow the general ANSI/UNIX convention of returning 0 on success and -1 on failure.

The list keeps track of both the head and tail elements, which greatly simplifies the `slInsert()` and `slAppend()` operations.

SList's can be used for creating stacks and queues as well as plain lists. The `slSelect()` function could be used to implement a sort of dynamic random-access array, which although functional, would be quite slow.

The Code

Figures 1 and 3 contain the source code and header file for the SList package. Figure 2 is a test program for the SList package. The user just enters strings, which get stored in a list. An empty line terminates the input, and the list contents are then printed to the screen. A trivial program, but it does demonstrate a function I often use lists for: string storage.

The code has been tested with Borland C++ 2.0, and there is nothing in the code that will prevent it being ported to any ANSI compiler under any operating system.

Conclusion

I've used lists in all sorts of programs. They're not the most efficient of data structures, but if speed and size are not critical, their pure simplicity can provide amazing power with no costly development time. Who says you can't get something for nothing?

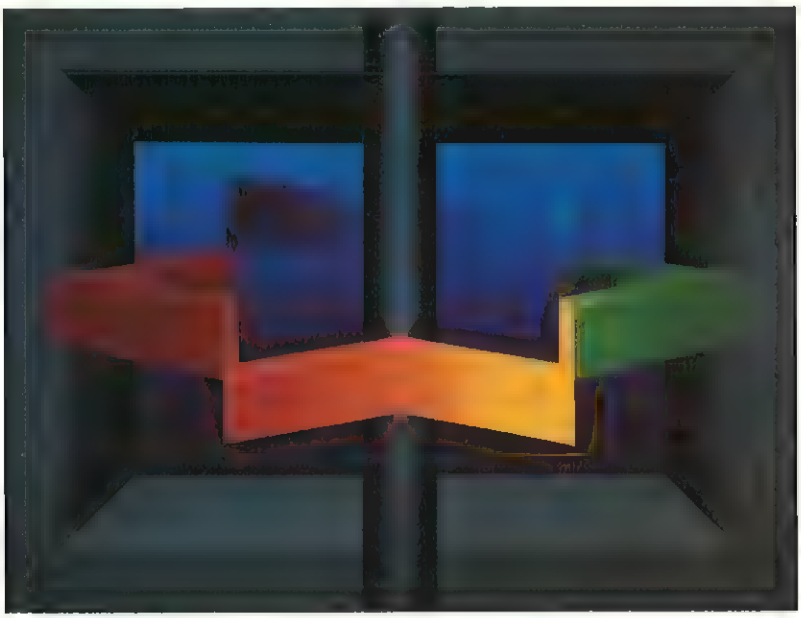
EXE

Michael John Marshall is a software engineer for X-ON Software, a City based software house. An American import, his interests beyond computers include science fiction, mythology, and origami. Michael can be reached on 071 522 0088, or michael@xon.co.uk

```
/*
** SLIST.C
** Linked List Package
** Michael J Marshall - X-ON Software
*/
#include <stdlib.h>
#include "slist.h"
/* Private Functions */
/* These functions can be rewritten to
   provide more sophisticated allocation
   schemes. My development version
   maintains a free node list.
*/
/* Returns a new list node,
   or NULL on error. */
static SLNode *newnode( void )
{
  return malloc( sizeof( SLNode ) );
}
/* Deletes a list node. */
static void freenode( SLNode *node )
{
  free( node );
}
/* Public Functions */
/* Creates a new, empty list.
   Returns the list, or NULL on error. */
SList *slNew( void )
{
  SList *list = malloc( sizeof( SList ));
  if ( list )
  {
    list->head = list->tail = NULL;
    list->curr = NULL;
    list->count = 0;
  }
  return list;
}
/* Destroys a list. If the del argument
   is not NULL, the provided function is
   used to destroy the list elements. */
void slFree( SList *list, free_t del )
{
  register SLNode *curr;
  if ( !list )
    return;
  for ( curr = list->head; curr; )
  {
    SLNode *next = curr->next;
    if ( del )
      /* delete the datum */
      del( curr->datum );
    /* delete the node */
    freenode( curr );
    curr = next;
  }
  free( list );
}
/* Similar to slFree(), except the list
   just emptied, not destroyed. */
void slClear( SList *list, free_t del )
{
  register SLNode *curr;
  if ( !list )
    return;
  for ( curr = list->head; curr; )
  {
    SLNode *next = curr->next;
    if ( del )
      /* delete the datum */
      del( curr->datum );
    /* delete the node */
    freenode( curr );
    curr = next;
  }
  /* reset the list */
  list->head = list->tail = NULL;
  list->curr = NULL;
  list->count = 0;
}
/* Adds a new element at the head of the
   list. Returns OK/FAIL. */
int slInsert( SList *list, void *datum )
{
  SLNode *node;
  if ( !list || !datum )
    return FAIL;
  if ( node = newnode() )
  {
    node->datum = datum;
    if ( list->head )
    {
      /* Insert into existing list */
      node->next = list->head;
      list->head = node;
    }
    else
    {
      /* First element in the list */
      list->head = list->tail = node;
      node->next = NULL;
    }
    list->count++;
    return OK;
  }
  return FAIL;
}
/* Adds a new element to the end of the
   list. Returns OK/FAIL. */
int slAppend( SList *list, void *datum )
{
  SLNode *node;
  if ( !list || !datum )
    return FAIL;
  if ( node = newnode() )
  {
    node->datum = datum;
    node->next = NULL;
    if ( list->tail )
    {
      /* Append to existing list */
      list->tail->next = node;
      list->tail = node;
    }
    else
    {
      /* First element in the list */
      list->head = list->tail = node;
    }
    list->count++;
    return OK;
  }
  return FAIL;
}
/* Retrieve the first element of the list
   and remove it. */
void *slGet( SList *list )
{
  if ( list && list->head )
  {
    void *datum;
    SLNode *node   list->head;
    list->count--;
    list->head = node->next;
    if ( !node->next )
      /* Removed only element in list */
      list->tail = NULL;
    datum = node->datum;
    freenode( node );
    return datum;
  }
  return NULL;
}
/* Used to iterate over a list
   non-destructively. If start is
   non-zero, the iteration is (re)started.
   Returns the next item in the
   list of NULL when the list has been
   exhausted.
*/
void *slIterate( SList *list, int start )
{
  if ( !list )
    return NULL;
  if ( start || !list->curr )
    /* Start iteration */
    list->curr = list->head;
  else if ( list->curr )
    /* Advance to next element */
    list->curr = list->curr->next;
  return ( list->curr ?
           list->curr->datum : NULL );
}
/* Searches the list for the first
   element that matches key, as defined
   by comparison function cmp, which
   should act similarly to strcmp().
   Returns the element if found, or NULL
   otherwise.
*/
void *slSearch( SList *list, void *key,
                compare_t cmp )
{
  register SLNode *curr;
  if ( !list || !key || !cmp )
    return NULL;
  for ( curr = list->head; curr;
        curr = curr->next )
  {
    if ( cmp( key, curr->datum ) == 0 )
      return curr->datum;
  }
  return NULL;
}
/* Removes the specified element from
   the list. Datum is the element in the
   list to delete. Returns OK if deleted,
   FAIL otherwise.
*/
int slDelete( SList *list, void *datum )
{
  register SLNode *curr;
  SLNode *prev = NULL;
  if ( !list || !datum || !list->head )
    return FAIL;
  for ( curr = list->head; curr;
        curr = curr->next )
  {
    if ( curr->datum == datum )
    {
      if ( !prev )
        /* Removing first element */
        list->head = curr->next;
      else
        prev->next = curr->next;
      if ( !curr->next )
        /* Removing last/only element */
        list->tail = prev;
      if ( list->curr == curr )
        /* Removing current element
                              - back up */
        list->curr = prev;
      freenode( curr );
      list->count--;
      return OK;
    }
    prev = curr;
  }
  return FAIL;
}
/* Returns the Nth item in the list
   (0 for first item). Returns element
   or NULL.
*/
void *slSelect( SList *list,
                register unsigned n )
{
  register SLNode *node;
  if ( !list || n >= list->count )
    return NULL;
  for ( node = list->head; n--;
        node = node->next )
    ;
  return node->datum;
}
/* If there is a current element,
   returns it, otherwise returns NULL. */
void *slCurr( SList *list )
{
  return ( list && list->curr ) ?
    list->curr->datum : NULL;
}
/* END SLIST.C */
```

Figure 3 - SLIST.C source file

Who put C++ to Work?



glockenspiel
class constructors

Glockenspiel C++

Glockenspiel C++ leads the field in object-oriented programming. It's the most effective and efficient implementation around. It conforms exactly to the AT&T C++ specification – and they don't come any closer than that! Portable too: it's on more platforms than any other C++ implementation.

Glockenspiel CommonView®

Glockenspiel CommonView is the leading C++ class library for developing Windows, PM and OSF/Motif applications. Logical, well structured and easy to use, CommonView is 3 to 5 times more productive than the API.

C++ Training & Consultancy

Glockenspiel C++ training courses reflect our experience as C++ developers. From design to development, *learn from the best!*
Our C++ consultancy services match your needs as you commit more resources to C++ and object-oriented development.

Further Information

GLOCKENSPIEL, 39 Lower Dominick Street, Dublin 1, Ireland. +353 (1) 733166. Fax +353 (1) 733034

■ **US: IMAGESOFT**, (516) 767-2233. Fax (516) 767-9067. **OASYS**, (617) 862-2002. Fax (617) 863-2633 ■ **UK: QA TRAINING LTD.**, (0285) 655888. Fax (0285) 650537.
■ **Italy: INFERENTIA**, (02) 26680568. Fax (02) 2364258. ■ **Sweden: LINSOFT**, (01) 3124780. Fax (01) 3152429. ■ **Germany, Switzerland, Austria: PSI**, (06021) 492-0. Fax (06021) 492-112
■ **Benelux: RIJNHAAVE**, +31 (71) 218121. Fax +31 (71) 216118

CIRCLE NO. 938

Patently obvious?

Richard Stallman, the man who started the League for Programming Freedom, has just completed a lecture tour of Britain. What does he think, and are we to take him seriously?

Suppose you invent a machine, and start manufacturing it. Any other company which can take it apart and find out how it works can start manufacturing it in competition with you. All your hard work involved in inventing and developing the machine now has to be shared with the other manufacturer, while they have contributed nothing to your effort. The situation is clearly unfair, so almost every country in the world operates a patent office. The idea of the patent office is that you can register your idea, and if the idea really is yours you will be granted a legal monopoly over the invention for a certain length of time. If anyone starts using your invention without your permission, you have the right to sue him.

"THERE ARE RESTRICTIONS ON WHAT YOU ARE ALLOWED TO PATENT"

There are restrictions on what you are allowed to patent. The law says that an application must describe a process or a machine which fulfils a specific purpose. Mental steps, mathematical processes, and computer programs are, therefore excluded. Certain types of machine are explicitly excluded, such as perpetual motion machines and anti-gravity machines, whether or not a working model can be demonstrated. Finally, the idea may not be public domain, it must not be obvious, and it must be the original property of the applicant. Because they are legal documents, lawyers must be employed by the applicant in order to cover the maximum possible territory with a single application. The situation in the US was, until recently, pretty much the same as it is here, until 1981 when *Diamonds v. Deibr* accidentally allowed software to be the subject of a patent. Since then a number of extremely questionable patents have been filed. Cadtrak, a company which exists solely to manage software patents, owns a patent which describes the use of exclusive-or to draw erasable cursors onto a computer screen. IBM owns patents which describe register optimisation in compilers, among others. The US patent office granted these patents, in spite of the fact that they seem to be obvious.

Naturally, this has generated a great deal of ill-will. The patent system is flawed. It is not hard to find all kinds of injustices, in this country as well as in the US. One can violate a patent that even an exhaustive search will not reveal (because the application is merely listed, not filed), and the resulting patent can be applied retroactively (one of the few retroactive laws we have). As the above cases show, the patent office themselves are not capable of assessing the (extremely technical) applications - when Quantel went to court in Britain with more than 60 patent claims (all of which were questionable in the terms of the act) many were amended at the trial before Spaceward was found in breach of them. And conferring an inventor with the right to sue an infringer is precious little use to most people - the costs involved prevent a private individual from suing a company of any size.

Now there is certainly a case to be made for reforming the patent system, but many people want to go further than that - they want software patents eradicated for good. I had real trouble with this concept - I couldn't see the difference between inventing in a workshop and inventing at a keyboard. After all, if someone spends years working on something, he has the right to expect a fair compensation for his effort. Richard Stallman disagrees. He sees a real difference between the two cases. Richard Stallman is the head man at the League for Programming Freedom, a pressure group set up to campaign against software patents. According to Stallman, writing in the ACM Communications (January 1992), software is different because a programmer's job is to invent.

I believe that software patents are very dangerous, and will do immense damage to the industry

Every programmer solves original problems in original ways, and many original solutions appear time and time again as they are re-invented by other programmers facing similar problems. He claims that a program of 50,000 lines is of about the same complexity as a car. The program can cost as little as a few tens of thousands of dollars to develop, and the car will cost many millions. They will both impact upon about the same number of patents. And here is the problem - if patent licensing costs about $100,000 in both cases (searches, lawyers costs, negotiation, and so on), the impact on the car will be negligible, but the impact on the program will be devastating.

I would like to add one further observation to this. When a patent is granted, it confers a monopoly on the applicant to use and develop the invention in all its forms. Practically all programmers, and most mathematicians hijack simple ideas with which to get a handle on the problem - the IBM patent about register optimisation specifically talks about colouring. The problem is, if a different solution can be shown to be equivalent to that described in a patent (either by changing the words, or by a mathematical transformation) then the new solution infringes the patent. Showing equivalence is not a trivial problem, and even if a proof could be established, its mathematical correctness will always be in some doubt. Court-rooms are not the place to argue maths!

I have now come round to Stallman's view. I believe, like him, that software patents are very dangerous, and will do immense damage to the industry. Although forbidding software patents will harm a few people, on a large scale the industry is inventing prolifically without any protection other than simple confidentiality, and we don't need them. We don't want them here, and the US should abandon them before the damage there is irrecoverable. The issues are too complex to allow them to be decided by ill-informed and barely applicable case law.

Postscript In the course of writing this item, I made contact with Stallman. Among the information he sent me was the following; 'Negotiations for the GATT (General Agreement on Tariffs and Trade) are continuing based on a working draft proposed in December as a compromise by the head of GATT. The draft would require all countries that accept the agreement to have patents that cover software techniques. It would also rule out all the ideas so far proposed to protect software from patents or to make the patent system bearable for software developers.'

The draft treaty covers all aspects of international trade, and it will be presented to the US Senate as a package deal. The result would be irresistible pressure to adopt the whole package. 'Thus, sweeping changes in US intellectual property law would be forced upon us and cast in stone, without any consideration by the House of Representatives, and with no opportunity for the Senate to consider them individually on their merits. At the same time, legal policy which Congress currently has the power to change would become frozen and unchangeable.'

GATT is intrinsically an international treaty, and I am advised that it is being considered here. The clause-by-clause commentary that Stallman sent me was truly frightening. Unless something is done soon, the damage will be irreparable.

EXE

Jules is an independent programmer and consultant who would be put out of business overnight if software patents were to be implemented in Europe. Chances are, so would you. He can be contacted on 0707 44185 or on CIX as `jules`. *The League for Programming Freedom can be contacted at 1 Kendall Square #143, PO Box 9171, Cambridge, MA 02139, phone 0101 617 243-4091 (from the UK), or at* `{league@prep.ai.mit.edu}`. *They are actively seeking new members.*

Contain Yourself!

There's lots of press about all the time spent (re)writing user interface code. But what about the other parts of your programs?

If half the bulk of a given project is user interface code, another 20 percent is data structures and their related functions. In particular, much time is spent developing data structures whose sole purpose in life is to contain collections of other data structures. In object-oriented circles such data structures are aptly called 'Containers'. Just as you can save time by using (or developing) a general purpose user interface library, you can also benefit from a general purpose container library.

Internally, containers can be implemented in many ways; as complicated as a wild variation of an undirected graph or as simple as an array, depending on the characteristics of the data being contained and how it will be accessed. Externally, the container can be thought of either in terms of its internal implementation, or in terms of its use characteristics (eg sometimes an unsorted array is called a 'Bag'). Examples of common container implementations are: Vector, SList (singly-linked), DList (doubly-linked) and BTree. Some common external views are: Set, Dictionary, Stack, Queue, Dequeue, and SortedCollection. Note that most of these external views could be internally implemented in several different ways (a Stack could be implemented as either a Vector or an SList, for example).

In a traditional programming language, the functions for manipulating data in a container (eg `Insert()`, `Get()`) would have to be rewritten for each different type of data to be stored in the container. C++ and other object-oriented languages allow you to write the container functions once and reuse them for several different types of objects. As a matter of fact, there are at least three different commonly used methods (in C++) for constructing reuseable containers. In the next few pages, I'll show you how each of these methods works and hopefully enlighten you to the relative merits of each.

Three Roads

The first type of container we'll discuss is the traditional '`void*`' container pioneered by Bjarne Stroustrup in the original *C++ Programming Language* and carried on in particular by the Zortech C++ Tools library. The basic concept of this method is that a) the same functions can work for different types of objects only if the size of the object is always the same, and b) as pointers are the same size no matter what they point to, storing pointers to objects in the container, rather than the objects themselves, allows reusing the same function code for different types of objects. Using `void*` circumvents most type-checking problems, as a `void*` is defined to be 'a pointer to anything' (quite different from a `void`, which is 'nothing')

The second method, '`Object*`' containers, is similar to the first, except that rather than having a container filled with `void*`, we make a container filled with `Object*`, where Object is a base class from which all other objects stored in a container must be derived. Borland C++'s Classlib implements this type of container.

The third method uses C++ 'templates'. This is similar to generic packages in Ada - you create a template with a place-holder where the actual object type will go. The template will later be used to generate container classes and their member functions as needed. As templates are fairly new to C++, many compilers don't yet implement them - BC++ 3.0 is one which does.

An Example

For the rest of this article, we'll be working with a simple example: a priority queue (call it `PQueue`). Data is inserted into a `PQueue` according to its priority (sorting order) relative to the items already on the `PQueue`. When data is requested from the `PQueue`, the item with highest priority is always given first. If two items have the same sort order, they are given back in 'First In First Out' (FIFO) order.

I have chosen `PQueue` because its implementation is relatively short, while still showing some of the problems that must be overcome in creating a reuseable container. In particular we need to compare two items to determine their relative priority (in `Insert()`), as well as needing to dispose of items (in `Clear()`).

Internally, I have implemented `PQueue` as a singly-linked list.

The test program for each method shows how to declare a `PQueue` of `int`s, and does a few `Insert()`s and `Get()`s to test it out.

void* Containers

Figure 1 shows `VPQueue`, an implementation of a `PQueue` using the '`void*`' method. First a `VNode` class is created to represent each node on the list - a `void*` for the data and a `VNode*` for the next item on the list. `VNode` will remain unchanged no matter what types of objects are stored on the `VPQueue`.

`VPQueue` contains a few twisters, however. The first problem encountered is that, in `Insert()` we must compare the priority of two objects to determine where the new object should be placed; unfortunately, `VPQueue::Insert()` itself cannot know how to compare these two objects, as it doesn't know their type. Virtual functions come to the rescue, with a `VPQueue::Greater()` function which can be redefined in later `VPQueue` descendants to cast the `void*` to the actual type, then do the compare (see `intVPQueue` in listing 1 for an example).

A similar problem occurs in `VPQueue::Clear()` when we try to delete objects as we remove them from the queue; calling `delete` itself would just dispose of the memory directly holding the object - any memory pointed to by the object would be left floating. This can also be solved by a virtual function, which we'll call `VPQueue::Destroy()`. `Destroy()` is redefined in later classes to cast its `void*` argument to the object's actual type, then call `delete`. This typecast makes sure that the object's own de-

structor is called, as well as the object's memory being freed. (My version of Zortech Tools just deletes the object as a `void*`, which leads to dangling memory allocations. So watch out!)

```
//              VPQUEUE.H
class VNode
  {  // one node on the Queue
  friend class VPQueue;
  void *data;
  VNode *next;
  VNode(void *d, VNode *n)
    { data = d; next = n; }
  };  // class VNode
class VPQueue
  {  // the Queue itself
  VNode *head;
public:
  void Insert(void *d);
  void *Get();
  virtual int Greater(void *a, void *b)
    { return 0; }
  virtual void Destroy(void *d)
    { delete d; }
  void Clear();
  VPQueue()     { head = 0; }
  ~VPQueue()    { Clear(); }
  };  // class VPQueue
//////////////////////////////////////
//          VPQUEUE.CPP
#include "vpqueue.h"
void VPQueue::Insert(void *d)
  {  // insert in sorted order
  if (!head || Greater(d, head->data))
    {  // first item
    head = new VNode(d,head);
    return;
    }
  VNode *n = head;
  while (n->next &&
      !Greater(d, n->next->data))
    n = n->next;
  n->next = new VNode(d,n->next);
  }  // VPQueue::Insert()
void *VPQueue::Get()
  {  // get highest priority item
  if (!head) return 0;
  void *ret = head->data;
  VNode *del = head;
  head = head->next;
  delete del;
  return ret;
  }  // VPQueue::Get()
void VPQueue::Clear()
  {  // clear entire queue
  while (head)
    {
    VNode *n = head;
    head = head->next;
    Destroy(n->data);
    delete n;
    }
  }  // VPQueue::Clear()
//////////////////////////////////////
//            VPQTEST.CPP
#include <iostream.h>
#include "vpqueue.h"
class intVPQueue : public VPQueue
  { // a priority queue of ints
public:
  void Insert(int *d)
    { VPQueue::Insert(d); }
  int *Get()
    { return (int *) VPQueue::Get(); }
  int Greater(void *a, void *b)
    { return *((int *) a) > *((int *) b); }
  void Destroy(void *d)
    { delete (int *) d; }
  };  // class intVPQueue
int main()
{
intVPQueue test;
test.Insert(new int(2));
test.Insert(new int(7));
test.Insert(new int(5));
test.Insert(new int(1));
int *p = test.Get();
while (p)
  {  // get and print all
  cout << *p << endl;
  delete p;
  p = test.Get();
  }
return 0;
}  // main()
```

Figure 1 - PQueue as a void container*

The main advantage of the `void*` method is that it is the easiest for a C programmer to understand. C jocks live for the reckless abandon of a 'pointer to anything', and the associated type-casts that go along with it. More seriously, while '`Object*`' containers can only contain objects derived from a specific base class (`Object`), a `void*` container can contain any type of object.

```
//////////////////////////////////////
//              OPQUEUE.H
//////////////////////////////////////
class Object
  {  // base class for items on Queue
public:
  virtual ~Object() {}
  virtual int Greater(Object *d)
    { return 0; }
  };  // class Object
class ONode
  {  // one node on the Queue
  friend class OPQueue;
  Object *data;
  ONode *next;
  ONode(Object *d, ONode *n)
    { data = d; next = n; }
  };  // class ONode
class OPQueue
  {
  ONode *head;
public:
  void Insert(Object *d);
  Object *Get();
  void Clear();
  OPQueue()     { head = 0; }
  ~OPQueue()    { Clear(); }
  };  // class OPQueue
//////////////////////////////////////
//          OPQUEUE.CPP
//////////////////////////////////////
#include "opqueue.h"
void OPQueue::Insert(Object *d)
  {  // insert in sorted order
  if (!head || d->Greater(head->data) )
    {  // first item
    head = new ONode(d,head);
    return;
    }
  ONode *n = head;
  while (n->next &&
      !d->Greater(n->next->data) )
    n = n->next;
  n->next = new ONode(d,n->next);
  }  // OPQueue::Insert()
Object *OPQueue::Get()
  {  // get highest priority item
  if (!head) return 0;
  Object *ret = head->data;
  ONode *del = head;
  head = head->next;
  delete del;
  return ret;
  }  // OPQueue::Get()
void OPQueue::Clear()
  {  // clear entire queue
  while (head)
    {
    ONode *n = head;
    head = head->next;
    delete n->data;
    delete n;
    }
  }  // OPQueue::Clear()
//////////////////////////////////////
//          OPQTEST.CPP
#include <iostream.h>
#include "opqueue.h"
class Integer : public Object
  {
public:
  int val;
  Integer(int v) { val = v; }
  int Greater(Object *d)
    { return val > ((Integer*) d)->val; }
  };  // class Integer
int main()
{
OPQueue test;
test.Insert(new Integer(2));
test.Insert(new Integer(7));
test.Insert(new Integer(5));
test.Insert(new Integer(1));
Integer *p = (Integer*) test.Get();
while (p)
  {  // get and print all
  cout << p->val << endl;
  delete p;
  p = (Integer*) test.Get();
  }
return 0;
}  // main()
```

Figure 2 - PQueue as an Object container*

```
//////////////////////////////////////
//              TPQUEUE.H
//////////////////////////////////////
template <class T> class Node
  {  // one node on the Queue
  friend class PQueue<T>;
  T *data;
  Node<T> *next;
  Node(T *d, Node *n)
    { data = d; next = n; }
  };  // class Node
template <class T> class PQueue
  {
  Node<T> *head;
public:
  void Insert(T *d);
  T *Get();
  void Clear();
  PQueue()     { head = 0; }
  ~PQueue()    { Clear(); }
  };  // class PQueue
template <class T>
 void PQueue<T>::Insert(T *d)
  {  // insert in sorted order
  if (!head || (*d > *(head->data)))
    {  // first item
    head = new Node<T>(d,head);
    return;
    }
  Node<T> *n = head;
  while ( n->next &&
      !(*d > *(n->next->data)))
    n = n->next;
  n->next = new Node<T>(d,n->next);
  };  // PQueue<T>::Insert()
template <class T> T *PQueue<T>::Get()
  {  // get highest priority item
  if (!head) return 0;
  T *ret = head->data;
  Node<T> *del = head;
  head = head->next;
  delete del;
  return ret;
  };  // PQueue<T>::Get()
template <class T>
 void PQueue<T>::Clear()
  {  // clear entire queue
  while (head)
    {
    Node<T> *n = head;
    head = head->next;
    delete n->data;
    delete n;
    }
  };  // PQueue<T>::Clear()
//////////////////////////////////////
//            TPQTEST.CPP
#include <iostream.h>
#include "tpqueue.h"
int main()
{
PQueue<int> test;
test.Insert(new int(2));
test.Insert(new int(7));
test.Insert(new int(5));
test.Insert(new int(1));
int *p = test.Get();
while (p)
  {  // get and print all
  cout << *p << endl;
  delete p;
  p = test.Get();
  }
return 0;
}  // main()
```

Figure 3 - PQueue as a template container

Imagine an object–oriented development tool for C++ that is more powerful than Windows or DOS specific tools. Imagine a transparent, single source code migration path between Microsoft Windows 3.X, DOS Graphics and DOS Text. Now turn your imagination loose with Zinc™ Interface Library™ 3.0.

- *Extensible, object-oriented, event-driven application framework.*
- *Fully optimized, single source code support for Microsoft Windows 3.X, DOS Graphics and DOS Text.*
- *Zinc Designer™–an interactive design tool for creating and saving application screens with platform-independent persistent objects.*
- *A comprehensive set of interface objects including windows, menus, scrollbars, icons, radio buttons, tool bars, lists, input fields (string, text, date, time, number), keyboard, mouse, help...*
- *Single package support for Borland C++, Microsoft C++ and Zortech C++ (Symantec) compilers.*

Call 1.800.638.8665 [IN EUROPE CALL: +44 (0)81 855 9918] and we'll send you a technical information packet detailing Zinc 3.0.

z i n c

Hardlock E-Y-E®– tying the hands of software pirates

The effective way to protect your software

FAST Electronic has made life a lot harder for software pirates. Hardlock E-Y-E was designed using cryptographic principles. It took the experience and know-how of Germany's number one in software protection and the leading edge technology of a US semiconductor company to create the ultimate software protection tool.

The technology programmers have at their fingertips

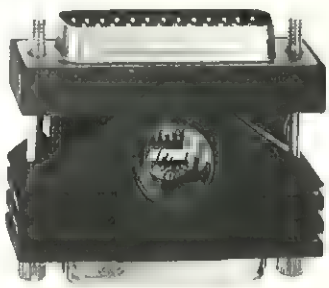
Hardlock E-Y-E is based on a custom chip and combines all the features that a programmer would expect from such a device: secure, algorithmic query routines and an optional non-volatile memory for custom configurations. With the Crypto-Programmer card from FAST you can program the algorithmic parameters and the memory within seconds. This unique card guarantees that no one else can burn Hardlock E-Y-Es with your codes. Linking Hardlock E-Y-E to your software is easy: you can either protect your .COM and .EXE files with the automatic encryption software HL-Crypt or integrate FAST's high level language routines into your source code.

Readily acceptable to your customers

Hardlock E-Y-E allows unlimited backup copies of the master floppy. The customer gets the device together with the software and plugs it into the parallel port between the printer and the PC. Daisy-chainability, outstanding reliability and the compact high tech design guarantee that your customers will accept Hardlock E-Y-E.

The benefits your management will appreciate

Hardlock E-Y-E can be programmed by the software house with the Crypto-Programmer card. This ensures optimum delivery schedules and stock flexibility. Revenues will go up as software piracy and multiple usage are prevented.



Hardlock E-Y-E programmable, algorithmic response and memory option – all in one

MAGNIFEYE

Order your demo unit today. Contact Magnifeye, 235-239 Walmer Road, Walmer Studio # 6, W 11 4 EY, Telephone 071 221 8024, Fax 071 792 3449.

Magnifeye is a subsiduary of Fast Electronic GmbH.

CIRCLE NO. 942

A `void*` container has some problems that just can't be overcome, though. The most obvious is that you must derive a new container class for each type of object you want to 'contain'.

This isn't very involved for a class like `VPQueue` (see VPQTEST.CPP in Figure 1) but becomes quite tedious for more complex classes like DList which have more member functions. Of course, you can follow the lead of Stroustrup (and Zortech) in using GENERIC.H macros to shorten derived class definitions to a single line, but I consider this blatant use of macros to be error prone, ugly, and in terribly low taste. Weren't inline functions created to reduce use of the preprocessor?

More importantly, the fact that the `Greater()` and `Destroy()` functions are part of `VPQueue`, and not of the object itself, means that all objects on the queue must use the same functions. This in turn implies that `void*` containers must be homogeneous - although different `VPQueues` can contain different types of objects, all objects on any one `VPQueue` must be of the same type. To make sure that nobody mistakenly tries to put a different type of object on a `VPQueue`, its descendants should redefine `Insert()`, as in `intVPQueue`, to specify the type of its argument.

The above problems point to a conceptual flaw in `void*` containers. A container should not need to (indeed, should not) know how to compare two objects it contains, nor should it know how to destroy those objects. That kind of detail should be known only by the objects themselves, otherwise there is a serious hole in the objects' encapsulation. Noting this flaw leads us to method two for designing containers.

Object* Containers

`Object*` containers remove the object comparison and destruction functions (and other possible functions not needed in this example) from the container class and put them with the object, where they rightfully belong. Figure 2 shows the `OPQueue` class which is implemented using the `Object*` method. The `Objects` themselves are pointed to by the data member of `ONode`.

You'll notice that `Greater()` is now a virtual member of `Object`. `Destroy()` no longer exists, as that duty is handled by a virtual destructor for `Objects`. If `Object`'s destructor is declared `virtual`, the destructors of its descendants are automatically made virtual, resulting in the proper destructor being called by `delete`, even if `delete` is given an `Object*` (instead of a pointer to the actual class of the object). The implementation of other functions is nearly identical to `VPQueue`.

Now the same type of container can be used to hold different types of object as long as they are all derived from `Object`. Even heterogeneous containers are possible - a single container can hold objects of different types (as long as the `Greater()` function knows how to compare objects of differing types). Of course, in return for this, we must derive a new class for each kind of object we want to store.

You'll notice, though, that in the example I must do a type-cast each time I call `Get()`. This is necessary because `Get()` returns an `Object*`, while we know that the `OPQueue` contains `Integers`. As `val` is declared in `Integer`, we can't access it through an `Object*`, we have to cast the `Object*` to `Integer*` first. This problem could also be solved in much the same way it was solved in `VPQueue` - by deriving a 'wrapper class' which did the type casting for us.

Like `void*` containers, `Object*` containers can never contain the data itself, just a pointer to the data. This is fine for large data items, but creates unproportionally large amounts of overhead when the container will be used just for `ints` or `chars`.

Template Containers

Figure 3 is an example of a Container designed using C++'s new 'template' feature. Where the previous two methods created functions for manipulating the objects in the container and used that same code for any type of object, templates merely create a 'stencil' of what the functions should look like. No code is generated until a container is declared (as '`PQueue<int> test`' in the example), when the actual class (`int` in our example) is substituted for the template class (`T`) and the container and its members are emitted. This is in many ways the same trick performed by GENERIC.H and the macro preprocessor, but cleaner and easier to understand.

Our previous problems of comparison and destruction are solved by the mere fact that when a `PQueue` is declared, new code is generated which creates exactly the correct comparison and destruction calls for the objects in question. Note that in order for `PQueue<xxx>` to compile correctly, the class `xxx` must have an `operator>()` function defined.

New code being generated for each different object type has good and bad implications. On the good side, the restriction that the data in containers always be the same size is lifted, meaning that you can create container templates which store objects rather than pointers to objects. This of course leads to faster executing code and more efficient use of data segment space. On the bad side, while source code remains short, .EXE size suffers.

Normal classes can have just the class definition in their header file, and their member functions compiled separately, but template class header files must contain the class definition AND the member functions. This allows the compiler to generate member functions from the templates when needed. Longer header files, of course, lead to longer compile times. I noticed something on the order of .01 second difference between the template version of my example and the other versions.

You can of course create heterogeneous containers based on template containers. Just make a base class like `Object`, and declare, for example, a `PQueue<Object>`. As descendants of `Object` will be different sizes, you must again stick to containers of `Object*`.

Which is Best?

This question can be answered with another question: Which do you have? Using an already existing container class is always easier than creating a new one. Keep in mind my warning about the Zortech Tools Library, and watch for similar problems in other `void*` container classes. If you want to make your own containers though...

Keeping in mind that templates can be used to create `Object*` or `void*` containers, and `void*` containers can be used to create `Object*` containers (with some acrobatics), it seems that templates are the way to go - even if you make an `Object*` container, do it with templates, that way you'll be able to reuse it for something else some day.

If your compiler doesn't have template capabilities, then use `Object*` methods. `void*` containers are merely a kludge for those who aren't yet totally comfortable with OOP. They may work, but they'll lead you down the wrong road.

EXE

Laine can be reached via the PC Tech BBS at (0101-612-345-4656, evenings, US time) or by post at: Bilkent University, Lojmanlari 3/9, 06533 Bilkent / Ankara, TURKEY.

Background tasks in Clipper

...or taking the 'wait' out of 'wait states'. Guy Smith shows how you can do-it-yourself.

Every now and again one of life's trivial details comes to the surface and stays long enough for a major revelation to occur. Times like these can be acutely embarrassing as you realise that the revelation is so simple and powerful there is no ready reason why it should have eluded you so long.

This text presents one of these simple ideas. The concept is simple and fundamental but a little thought can show how powerful it can be. The idea may well change your approach to programming and your attitude to Clipper.

The Idea

The idea is simply this : All our applications use many wait states. These wait states need not go to waste; they can be used for our own purposes.

The simplest of wait states is `Inkey(<nSecs>)` and this is the one we will attack. This wait state occurs in two of the most common wait states in an application: the get system and browsing. We will return to the other wait states later.

We need to replace this wait state, `Inkey(<nSecs>)`, with one under our own control, `BTskInkey(<nSecs>)`. The easiest solution here is to copy STD.CH to MYSTD.CH and add this line:

```
#translate Inkey(<nSecs>) =>
 BTskInkey(<nSecs>)
```

set the CLIPPERCMD environment variable:

```
SET CLIPPERCMD=/UMYSTD.CH
```

then recompile all PRGs. All references to `Inkey()` will change to `BTskInkey()`.

BTskInkey()

`BTskInkey()` fulfils the same purpose as `Inkey()` with the addition that it also processes background tasks.

Background tasks are tasks which can occur during a wait state. Before providing a few examples let's discuss exactly how these tasks will be managed and processed.

`BTskInkey()` counts down the number of seconds (or forever if seconds is 0) not by using `Inkey(<nSecs>)` but by marking the start time and counting down from there. See Figure 1.

In this way control is retained by `BTskInkey()`. Into the loop we simply need to add a function to process the background tasks, `BTskEval()`.

Managing tasks

BTSK, shown in Figure 2, contains all the functions which are needed to manage the background tasks. The external static `aTasks` is an array where each element is an array describing a background task. The background task consists of a task handle and the task code block. It is important to note that the task list is not a stack as tasks can complete at different times and may, therefore, remove themselves from the task list in a seemingly random order. Hence the task handle returned is a uniquely generated number identifying the task rather than the element number of the array as the element number is not a constant.

The `BTskInkey()` function listed in Figure 2 includes a number of simple refinements:

The number of the last task executed is saved in a static variable so that the next wait state starts where the last finished and an even distribution is maintained.

If `BTskInkey()` runs out of tasks to execute it calls a genuine `Inkey()` to enable Clipper's VMM to perform garbage collection.

If the return result of the task is a number it is taken as a keypress and returned as the actual key pressed. Thus key filtering and substitution can take place.

`BTskInkey()` passes a parameter to each task indicating how many times it has been called by the current wait state. This information is required by some background tasks.

Timeout

The most simple of background tasks watch for a timeout situation. Many programmers will already have modified GETSYS.PRG with this aim but such changes are necessarily limited to `GET/READ` and must be dutifully made to each reader function. Often it is useful to operate an application-wide timeout and either re-demand the user's password or log the user right out of the system.

The two functions required for this are shown in Figure 3. The functions shown allow for just a single timeout but it would be relatively easy to modify them for more than one timeout task.

The `BTOut()` function maintains a start point using a static variable which is initialised the first time it is called for each wait state (ie when `nCalls` is 1). If the difference between the start point and the current time is greater than the timeout seconds a timeout occurs.

Clock

The number of uses for a background task are numerable and varied. Maintaining an on-screen clock would be simplicity itself. All that need be done is to add a task which printed `TIME()` at a given location on the screen. However the clock would only be updated during wait states so it would give better results if just hours and minutes and no seconds were shown. A simple variation on this idea would be a stopwatch which displayed time elapsed.

Perhaps the most dramatic and simple uses for a timeout would be a browse refresh or

```
function BTskInkey(nSecs)
local nStartSecs:=seconds()
local nKey:=inkey()
do while
(nSecs==0 .or. seconds()-nStartSecs<=nSecs)
.and. nKey==0
   ...
   ...
   ...
  nKey:=inkey()
enddo
return(nKey)
```

Figure 1 - Inkey() written in Clipper

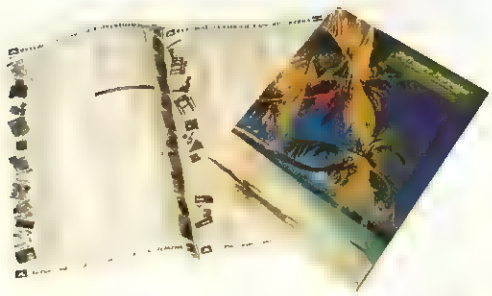
A WORLD OF

The widest range of software is only a phone call away

FAST - Federation Against Software Theft says £300m was lost by U.K software publishers alone through piracy in the year to March 1990. Software Paradise support and encourage the ethical use of software. To report copyright violations call FAST on (0628) 660377

The Widest Selection of Applications and Development Tools Anywhere

Place Orders Free on 0800-378-873



MasterCard VISA Access

Just Call Us - It's That Easy!

Why struggle with traffic? Why fight crowds? Just pick up the phone and say hello to the convenient way to buy ALL your PC software. Software Paradise Expressline is the fast, new distribution system for all of today's popular titles. Choose from the widest range of software available from any single source, all listed in our FREE 100-page Buyers' Guide. Place your orders FREE and the software will be despatched to you directly.

Software Paradise has supplied 100,000's of customers since 1986 and has fast become the U.K's leading software supplier. Our service is both convenient and affordable with up to 50% off manufacturers RRP's and the widest range anywhere!

Call us today and discover a World of Software!

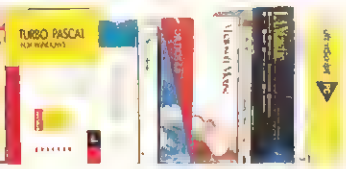
Software Paradise®

MORE SOFTWARE THAN ANYONE ELSE!

Call 0222 887521 • Fax 0222 862209

SOFTWARE To Go!

Over 15,000 Software Products for DOS, Windows™, OS/2®, Mac & UNIX®



* If you find an identical software product advertised in the national computer press by another mail-order company based in the U.K at a legitimate, current price - tell us at the time of placing your order and we will happily BEAT that price and SAVE YOU MONEY! Join the 1000's of customers who have already benefited!

Don't forget - We'll BEAT Any Nationally Advertised Price!*

WHAT MAKES US DIFFERENT

- ✔ FREEPHONE 0800-378-873 Orders Only
- ✔ FREEPOST Mailing Address
- ✔ 24-Hour FAX Orderlines
- ✔ Up to 50% Discount On Over 15,000 Software Titles
- ✔ Pre-Sales Consultation
- ✔ FREE Post-Sales Technical Support
- ✔ Bulletin Board 0222 865551 (2400,8,N,1)
- ✔ Compuserve Mailbox 100015, 3570
- ✔ Unbeatable Prices*
- ✔ Next Day Delivery
- ✔ Huge Software Inventory
- ✔ FREE Literature Department
- ✔ Friendly & Knowledgeable Sales Staff
- ✔ VISA, ACCESS & Mastercard Accepted
- ✔ No Credit Card Charges Until Shipping
- ✔ Corporate P.O's Welcome
- ✔ Educational & Volume Discounts
- ✔ Latest Versions Guaranteed with Full Documentation & Manufacturers Warranties
- ✔ Over 6 Years Experience Providing the Very Best in U.K and U.S Software
- ✔ Stable & Secure Company with Impressive Track Record & Enviable Balance Sheet

Products shown in this advertisement represent only a small sample of our full range. Please specify Media Size when ordering. Goods are not offered on a trial basis. All our prices exclude Carriage & VAT.

Software Paradise, Avenue House, King Edward Avenue, Caerphilly, Mid Glamorgan, CF8 1HE

FREEPOST -TODAY-
NO STAMP NEEDED

☐ Please rush me my **FREE** copy of *'The Discerning Persons Guide to Software'*

NAME

COMPANY

ADDRESS

TEL FAX

Mail to: Software Paradise, FREEPOST, Caerphilly CF8 1ZZ

EXE 5/92

CIRCLE NO. 946

CLIPPER ADD-ON SOFTWARE

QBS Software Limited specialise in providing and supporting the best add-on libraries and utilities for Clipper developers throughout Europe. We stock a range of libraries, utilities and now books. We will also source software that is not currently on our product list. For information on products please contact our technical support team by phone, fax or via our bulletin board.

We publish a quarterly news booklet, QBS Software News, which is sent free to all registered users. It is packed with product information, news on new releases, bug fixes and articles by renowned Clipper gurus and authors.

FUNCTION LIBRARIES

BabelFish	Paradox Database Driver
Bar Code Library	Produces most bar codes. Supports Epson, HP, Postscript printers
CL Text	Linkable, configurable object orientated word-processor
ClipWKS	Read/Write Lotus/Quattro Pro
dbXstore for GFORCE	Save and restore GFORCE graphics screens compressed
Dr Switch ASE	Create RAM resident applications
Fast Text Search	Advanced Text Search Technology
Flexfile	Variable length fields and .DBT replacer
FUNCky 1.52	Powerful Function Library with over 400 functions
GFORCE	Fast graphical interface Library with Screen Editor and graphical
Grumpfish Library	Friendly Function Library
Grumpfish Menu	Smart Menu Systems
Netlib	Novel and Netbios Networking Library
Overlay()	Memory Roll Out Library
PegLib	Linkable Pegasus read/post functions
Scripton	Comprehensive PostScript Library available for Clipper and C
SilverClip	Professional Communications (Includes full source)
SilverComm	Communications Library with full control of serial devices
SilverFox	Communications Library for FoxPro II
SilverPaint	Graphics Library for Clipper/FoxPro II
SubNtx()	Fast Filtering Utility
The Engine	Linkable Spell Checker

UTILITIES

9K for Clipper	Version Control Without Tears
Blinker	The Dynamic Overlay Linker - Now multi-lingual
Clipper Technical Guides Vol. I	TBrowse, Get System, Principles plus sample disk
Ed for Dos	The Programmer's Editor
Ed for Windows	The New Windows Version
Embark	Installation and Management
Expert Help	Drop-in replacement for Norton Guides
R & R	Relational Report Writer for Clipper, FoxPro, dBase and Paradox
R&R Code Generator	Create programs from R & R reports
SpellCode	Jake Geller's Spell Checker
ZipitUp	

CLIENT SERVER

Biton	
ClipSQL	SQL Server/Sybase
ExtendBase	Client Server with .DBF files
RaSQL/B	Access Btrieve files
RaSQL/X	For Novell Netware 386
SQLBase	Access Gupta SQLBase

OOPS TECHNOLOGY

BaseClass	Clipper 5 Object Library from Chydale
dbXstore	Save SuperClass Objects
StarClass	User Interface Class(y) Library
SuperClass II	Design your own classes
ViewIt	User Interface Class Library

90 days free technical support by phone or fax. 30 day money back guarantee

For further information, free demo software, prices and how to order please contact:

QBS Software Limited, 10 Barley Mow Passage, London W4 4PH
Tel: +44 81-994 4842 Fax: 44 81-994 3441 BBS: +44 81-747 1979

READ refresh. A background task could be set which, after every 30 seconds or so, sent the `refreshAll()` message to a given `TBrowse` object. Alternatively the background task could refresh a READ every 30 seconds or so, or refresh the screen only when the data is changed by another user.

Keyboard scripts

A task can effectively control the keyboard. On this basis two simple functions, `KbdRecord()` and `KbdPlayBack()`, allow the recording and playback of keyboard macros including keystrokes which are outside the capabilities of the KEYBOARD command. A call to `KbdRecord()` must be included in `BTskInkey()` immediately before the `Return` statement to enable keystrokes to be recorded. `KbdRecord()` and `KbdPlayBack()` are shown in Figure 4. A simple variation on the keyboard tasks and timeout tasks would be to lock the keyboard after a period of inactivity.

A more involved variation on the keyboard tasks would be to provide an interactive tutorial or demo script which actually ran the program concerned and taught or demonstrated in a live environment.

Popup calendars and popup calculators could be set to create playback scripts in order to paste dates or numbers into the application proper.

Messaging

Inter-terminal messaging is a great tool for easing user frustration and alienation. A function, say `ITMInstall()`, installs a message detector, `ITMWatchDog()`, as a background task and opens a shared table of messages, ITM.DBF.

Each new message is given a unique message number, a *From* user ID and a *To* user ID. The *To* user ID could be a specific user or *ALL* to broadcast a message to all users.

The `ITMWatchDog()` function needs only to softseek for a message number greater than the last message read which is addressed to the user or to *ALL*. If such a message is found `ITMWatchDog()` sounds the bell and displays a message. In this way the mail need not interfere with the running of the application.

A hot key is set to `ITMManager()` which would allow the user to create, read, reread and delete mail.

Other applications

There are several simple variations on the messaging theme. An alarm clock background task is an obvious variation. How-

```
* BTSK.PRG
static aTasks:={}

function BTskInkey(nSecs)
* Purpose :- to operate almost identically to the standard
* Clipper INKEY() function with the exception that SET KEYs are
* processed properly and background tasks are processed

local bkeyblock,nKey
do while .t.
  nKey:=iif(valtype(nSecs)=='N' .and. nSecs>=0 ,BInkey(nSecs),
inkey())
  if (bkeyblock:=setkey(nKey))==NIL
    exit
  else
    eval(bKeyBlock, procname(2), procline(2), '')
  endif
enddo
return(nKey)
*
* eofunc BTskInkey()
*

static function BInkey(nSecs)
* Purpose :- provide an inkey for a number of seconds
* 'nBTsk' is the task number
static nBTsk:=1
local nKey:=inkey()
local nStartSecs:=seconds()
local nCalls:=0
local nStartingTask:=nBTsk
local xResult

do while (nSecs==0 .or. seconds()-nStartSecs<=nSecs) .and. nKey==0
  if BTskCount()>0
    * there are background tasks to be processed
    if nBTsk > BTskCount()
      * go back to the beginning of the list
      nBTsk:=1
    endif

    if nBTsk==nStartingTask
      * increment the number of times the task has been called
      * within the current wait state
      nCalls++
    endif

    * evaluate the background task and increment the task pointer
    xResult:=BTskEval(BTskNum2Handle(nBTsk++), nCalls)
    if valtype(xResult)=='N'
      * use the subtituted value as the key pressed
      nKey:=xResult
      exit
    else
      nKey:=inkey()
    endif

  else
    * there are no background tasks so give Clipper's VMM a chance
    nKey:=inkey(nSecs)

  endif

enddo
return(nKey)
*
* eofunc BInkey()
*
```

```
function BTskCount()
* Purpose :- to return the number of tasks in the task list

return(len(aTasks))
*
* eofunc BTskCount()
*

function BTskNew(bTask)
* Purpose :- to add a new task to the task list and return
* the task handle
* 'nTskHandle' is a unique handle by which a task is known

static nTskHandle:=0
if valtype(bTask)=='B'
  aadd(aTasks, {++nTskHandle, bTask})
endif
return(nTskHandle)
*
* eofunc BTskNew()
*

function BTskNum2Handle(nElem)
* Purpose :- to return the task handle at the position 'nElem'
local nHandle:=0
if valtype(nElem)=='N' .and. nElem>0 .and. nElem<=len(aTasks)
  nHandle:=aTasks[nElem, 1]
endif
return(nHandle)
*
* eofunc BTskNum2Handle()
*

function BTskHandle2Num(nHandle)
* Purpose :- to return position of the task handle
local nElem:=0
if valtype(nHandle)=='N'
  nElem:=ascan(aTasks, {|aTask| aTask[1]==nHandle}) endif
return(nElem)
*
* eofunc BTskHandle2Num()
*

function BTskEval(nHandle, nCalls)
* Purpose :- to evaluate the task identified by nHandle
local nElem, xResult

if (nElem:=BTskHandle2Num(nHandle))>0
  * nHandle is valid
  xResult:=eval(aTasks[nElem, 2], nCalls)
endif
return(xResult)
*
* eofunc BTskEval()
*

function BTskDel(nHandle)
* Purpose :- to remove a task from the task list
local nElem

if (nElem:=BTskHandle2Num(nHandle))>0
  * nHandle is valid
  aDel(aTasks, nElem)
  aSize(aTasks, len(aTasks)-1)
endif
return(NIL)
*
* eofunc BTskDel()
*
```

Figure 2 - BTSK.PRG (Background Task Management)

ever, the alarm clock need not be limited to a simple message. By using a code block instead of a character string for the message the application can be made to perform certain tasks at specific times of the day.

The messaging system need not be restricted to sending messages between the users. Inter-application messaging where the applications themselves generate messages to send between themselves of which the user is unaware and requires no knowledge or action is quite feasible. For example if a lock failed the application requiring the lock could broadcast a message to all other workstations asking who owns the lock.

Taking this idea a step further it follows that if the application can send itself internal messages telling it to do something (ie timed tasks) then a system controller could send inter-terminal messages to tell specific or all workstations to perform certain tasks. Thus remote logoff and, indeed, remote control of any nature is possible right down to the control of the keyboard.

```
function BTOutNew(nSecs, bTimeout)
* Purpose :- to install a new timeout
*
return (BTskNew({|nCalls|
  BTOut(nCalls, nSecs, bTimeout)}))

function BTOut(nCalls, nSecs, bTimeout)
* Purpose :- to look for a
* timeout situation

static nStartSecs
local nKey
do case
   case nCalls==1
     * this is the first time this
     * function has been called for the
     * current wait state so initialise
     * the starting point
     nStartSecs:=seconds()
   case seconds() nStartSecs>nSecs
     * timed out
     if valtype(bTimeout)=='B'
       nKey:=eval(bTimeout)
     endif
     if .not. valtype(nKey)=='N'
       nKey:=0
     endif
endcase
return(nKey)
```

Figure 3 - Timeout Background Task

Problems

As I mentioned at the beginning of the article, `Inkey()` on which the whole idea is based is not the only wait state. Other wait states for which we do not have control include: `MemoEdit()`, `aChoice()`, `Alert()` and `@ PROMPT / MENU TO`. We will ignore `WAIT` and `dbEdit()`.

The most obvious solution is to rewrite each of these in Clipper solely for the benefit of using our own `Inkey()` function. For `Alert()` this is not so difficult but it is not practical for `MemoEdit()` without the aid of a memo class library.

The solution for `MemoEdit()` is not aesthetically pleasing. It involves forcing the wait state into the UDF called by `Memo-`

```
#define KBD_BEGINRECORD         1
#define KBD_ENDRECORD           2
#define K_SEQUENCE_TERMINATOR  K_CTRL_F1
static aKeys:={}
function KbdRecord(nKey, nNewMode)
static nMode:=KBD_ENDRECORD
if nKey==K_SEQUENCE_TERMINATOR
  * user terminated record sequence
  * using Ctrl-F1
  nNewMode:=KBD_ENDRECORD
endif
if valtype(nNewMode)=='N'
  * new mode is being set
  nMode:=nNewMode
  if nMode==KBD_BEGINRECORD
    aKeys:={}
  endif
endif
if nMode==KBD_BEGINRECORD
  aadd(aKeys, nKey)
endif
return(aKeys)
function KbdPlayBack()
local nKey
if len(aKeys)>0
  nKey:=aTail(aKeys)
  aSize(aKeys, len(aKeys)-1)
endif
return(nKey)
```

Figure 4 - Keyboard Record and Playback

`Edit()`. The UDF would contain a wait state which would supply keys to `MemoEdit()` and stuff an additional keystroke into the keyboard buffer in order to return control to the UDF. Not an elegant solution but it is a solution nonetheless.

An additional problem is the use of `LastKey()`. Many of the tasks outlined in this article rely on being able to substitute a keyboard value (timeout, keyboard scripts, remote keyboard control). `LastKey()` returns the last key physically pressed which may or may not be the last keystroke supplied to the application.

What is required is a replacement for `LastKey()`, `BTskLastKey()`, which can be integrated in the same way as for `Inkey()`. `BTskLastKey()` simply returns the last key supplied by `BTskInkey()`. The code for `BTskLastKey()` is left to the reader.

```
function BTVMMNew()
* Purpose :- to install a garbage
* collection background task
return(BTskNew({|nCalls| BTVMM(nCalls)}))
function BTVMM(nCalls)
* Purpose :- to perform garbage collection
* at the beginning of each wait state
if nCalls==1
  memory(-1)
endif
return(NIL)
```

Figure 5 - Memory Garbage Collection

The last problem concerns Clipper's VMM. As a genuine wait state is virtually never entered into, VMM's garbage collection will only be performed on an 'as needed' basis thereby slowing down the application. A simple solution to this would be to set up a background task which mirrors Clipper's normal use of VMM in `Inkey(0)`. To do this we need to know that one of the undocumented parameters to the `Memory()` function can be -1. `Memory(-1)` performs garbage collection immediately. Thus the background task in Figure 5 performs garbage collection each time a wait state is started.

Conclusion

Many of the techniques described here are available from networking libraries, other add-ons and additional software. However, rather than being a poor man's substitute this method has advantages beyond other solutions. The solution is generic and will work on any network, and most importantly, it is integrated into the application. The approach is analogous to the arguments for using classes and methods over variables and functions.

Of course, in the long run Nantucket will provide more elegant solutions to many of these problems when it includes its event handler. Unfortunately it is not possible to create a fully functional event handler using Clipper 5.01 and until then this solution is simple to implement and provides many features previously described as impossible or too difficult to implement.

EXE

Guy Smith is a freelance writer and has developed in Clipper, amongst others, since Clipper Winter '85. He is the author of the Capella Library, an application development library for Clipper 5. Guy can be contacted on CompuServe at 100014,1724 or on CIX as `smithg`.

The code presented in this article plus a full implementation of a background print manager is available on disk. Send a blank formatted floppy disk to the Editor, following the instructions given on Page 1, column 1. Mark your envelopes 'BTSK'.

Develop Complex Client-Server Solutions - FAST

NEW

FastStart. From the UK's leading technical consultancy and training organisation. You need to create ever more complex IT Solutions in less time, and using all the latest technology, such as Client-Server, Graphical User Interfaces and Object-Oriented Programming. FastStart is your solution, a package comprising expert consultancy, specialised training and rapid application development using the acclaimed Enfin development tool.

QA

THE CONSULTANCY

1 Our consultants have considerable experience in building GUI Clients working with powerful servers. FastStart consultancy will help specify, implement and deliver the solution, transferring their skills to your team in the process.

THE DEVELOPMENT TOOL

2 The heart of rapid application development is the development tool, and Enfin is the best. For Windows or OS/2, nothing comes close to Enfin's power, flexibility and advanced Toolset, including GUI and Database tools. Enfin is fully Object-Oriented, so solutions you develop today can be re-used tomorrow.

ENFIN

THE TRAINING

3 QA is the leader in technical training, and QA will back your FastStart with a carefully chosen training programme. With training in the operating environment, in Enfin, in advanced programming skills, in supporting your application when it's delivered, in fact whatever's required.

Yes ☐ Please send me full details of how to get a FastStart to developing complex client-server solutions for my business.

NAME

POSITION

COMPANY

ADDRESS

POSTCODE

TELEPHONE

FSEXE1

Call (0285) 655888 now, or mail to QA, Cecily Hill Castle, Cirencester, Gloucestershire, GL7 2EF, UK.

CHARGE NO. 954

QA

FASTSTART

QA, Cecily Hill Castle, Cirencester, Gloucestershire, GL7 2EF, UK. TEL. (0285) 655888

PC Security from Microft Technology.

MENUGEN

MENUGEN provides a password protected menu interface. The system manager has complete control of what each user is permitted to do. Access to the DOS prompt can be prevented. MENUGEN has no memory overhead. **£48** + **VAT.**

CLAM

CLAM provides TOTAL security. It incorporates MENUGEN and D-LOCK. In addition it provides a number of other facilities, the most important of which is file encryption – the only way to provide a high level of security on a micro. **£148** + **VAT.**

D-LOCK

D-LOCK prevents access to the hard disk after booting from diskette. **£38** + **VAT.**

S-LOCK

S-LOCK clears the screen and locks the keyboard after a specified period of inactivity or on request. The screen is restored and the keyboard unlocked only when the correct password is input. **£38** + **VAT.**

Security Consultancy.

A professional computer security consultant can help you determine the steps required to protect your computer information and systems. The options offered range from a fixed price risk analysis with action advice to a full security audit.

Microft Technology Limited
The Old Powerhouse, Kew Gardens Station
Kew, Surrey TW9 3PS.
Telephone: 081-948 8255

Anti-virus software is also available. Corporate licences are available for all products.

Terminals

This month Peter Collinson clears the muddy waters of UNIX terminal I/O.

In October of last year, *.EXE* contained an article by Ben Thompson about his struggles with the UNIX terminal interface and how he overcame them. Ever since then, I have wanted to complete the picture, as it were, to sketch in how things work and why this seemingly over-complex structure is in place.

Life is easier in a single-user system. There is only one keyboard to deal with at any one time. When a program gets a character from the keyboard, that's all the CPU is doing. It can happily sit in a tight loop polling away while it waits for the user to put down that cup of coffee and do something. When the program finally has a character and wants to echo it back to the user then there is only one place to put it, only one screen.

There were small single-user systems around 25 years ago when UNIX was created. Machines then were orders of magnitude more expensive. The fundamental ideas of time sharing sprang from the desire of people to use the expensive hardware to its fullest. It's perhaps curious how these multi-processing systems have become the cornerstone of many people's single-user workstation environment.

Multi-user systems

The designers of UNIX were making a multi-user, multi-processing system. There would be more than one terminal attached to the machine so programs that talk to the terminal must not have particular hardware addresses burnt in. All terminal operations must take place independent from the actual device that the user is using.

The aim is to provide complete device-independent operation to all programs. The model of device-independence that UNIX uses is the byte stream. Many UNIX programs read streams of data on their input channels and expect to generate streams for output. We want all these programs not to have to do anything special when talking to a terminal, or any other device.

This work is done by the device driver. User processes will issue `read` or `write` system calls to move data in and out of the kernel. Because the file descriptor used in these system calls is a device, eventually the process will find itself running the code in the device driver.

If you alter terminal settings, you must never rely on the shell resetting things after you

You should think of the device driver as having two halves. The 'top level' provides an interface to the processes, moving data between the kernel and the process address space. The 'bottom level' provides an interface to the device. It moves data between the device and the 'top level' in the kernel. It is generally interrupt driven.

The top and bottom levels are two asynchronous code paths or threads of control. The top level is called directly by the user process. When the user process issues the appropriate I/O system calls, the top level becomes part of the user's code path, part of his thread. The bottom level is called by the interrupt system on the machine and runs independent from any user process. The two levels communicate by a data buffer.

The top level will *block* when it has nothing to do. A process that is writing will block when there is not enough space in the output buffers to put yet more characters from the user's address space. Once blocked, the top level code will not restart until the kernel issues a wakeup call.

There is a system load imposed by the context switch when we stop and start processes. We want each section of the system to do as much as it can before giving up the processor. We want the top level code to move as much data as it can between the user process and the kernel. It can then be put to sleep for some lengthy period of time while the kernel gets on with shipping data to or from the device. When it has done, the kernel will wake up the top level to move more data.

Terminal reads

A terminal keyboard driver that is reading characters from the user in the normal mode (sometimes called *cooked* mode) operates in units of lines. It will store the user's data until *return* is hit. Only then will a blocked top level be woken to move the lines from the kernel internal buffers into the process address space. To minimise top level restart, the kernel takes on the load of keeping the user happy by echoing any input and doing any line editing that may be needed.

To see this in practice, let's follow through the sequence of events that happen when we read characters from the terminal. We start at the point where the user process issues a `read` system call that will eventually call the device interface.

There may be characters ready in the buffer for that terminal. If there are, they are moved into the process and that's that. The independent bottom half of the device driver can run at any time. UNIX permits type-ahead, characters can be entered when there is no process available to read them. The data lurks in the terminal buffer until some process comes along. If the buffers get too full, then inbound characters are simply discarded.

If there are no characters available in the buffers for that terminal when the process issues its `read` system call, then nothing can be done. The process is marked as waiting for input from the terminal and blocks until the user types something.

An interrupt will be raised when the keyboard is struck. The kernel catches the interrupt and dives into the device driver bottom level. This reads the character from the keyboard and resets things so that the next input character will set off another interrupt.

The kernel is holding a character and has some choices depending on how the interface is configured. In the normal case, the action that is taken depends on the character that has been entered. Most characters are simply placed in a buffer and retained in the kernel until a *return* is entered. They are also put onto the output queue so they find their way to the screen as a character echo.

Other characters are used to permit the user to interact with controlling functions within the terminal driver. The character could be one concerned with editing the input line.

The character could be one of those that results in a signal being sent to the processes that are using the terminal. The kernel delves into a table of user settable characters to see if the character should cause a signal eg ^C. Some systems permit the user process to request that a signal is sent on the receipt of any character. If any of these signals is to be sent, then the kernel will post the signal to the appropriate processes.

When the user strikes *return*, it tells the kernel that the user wants to send any stored characters to the top level. What happens here is system dependent. Some systems will move character buffers up through layers of code. Some early systems were worried about the time spent at interrupt level and did not edit the data buffer as the user deleted characters. Instead a marker was placed in the buffer and the processing was done at the top level. The 'real' data was created by making a 'canonical' buffer.

Whatever the mechanism, the top level is woken up and moves the data into the address space of the process. The `read` system call will return with the number of characters that were moved. All programmers are used to dealing with this. All programs are written to expect that `read` may return less than the requested number of bytes.

Controlling the interface

Normal mode of working may not be what the user process wants. The complexity of the terminal device driver springs from the desire to allow the process to select only the parts of the kernel processing that it needs. These days in UNIX systems, this is done by `ioctl` calls. Original UNIX systems used the `stty` system call to set information and the `gtty` call to retrieve it. One original name is now seen as a command that gets and sets terminal information: `stty`.

The aim is to provide complete device-independent operation to all programs

POSIX has defined a new set of routines that are used to get and set terminal information. These routines move a C structure called `termios` in and out of the kernel. Setting bits and values in the structure will alter the way that the interface works. The structure is a clone of the System V way of doing things and will look a little like:

```
struct termios {
    unsigned short c_iflag;
    unsigned short c_oflag;
    unsigned short c_cflag;
    unsigned short c_lflag;
    unsigned char c_cc[NCCS];
};
```

Things may vary on your machine. The variable `c_iflag` has bits that control basic actions on input. The variable `c_oflag` controls output translations and character delays. The hardware control of the terminal is determined by `c_cflag`. This is where the baud rate is set and parity selected.

In older systems, a single bit in the terminal control structure was used to show that the terminal was to work in raw mode.

The more modern structure splits the control of the various aspects of processing into many more pieces. This means that you must use a combination of bits to emulate the old raw mode.

Forwarding from the bottom to the top level is controllable too. First you turn off any processing of editing characters by clearing the ICANON bit in the `c_lmode` word. Then two bytes in the `c_cc` vector become variables that control top level wakeup. On my Sun, the variables are `c_cc[VMIN]` and `c_cc[VTIME]`. Your implementation might be different and I call these TIME and MIN in the following discussion. The MIN variable represents the minimum number of characters that should be received before the wakeup is actioned. The TIME variable is a timer of 0.10 seconds that is used to timeout bursty or short transmissions.

Setting MIN to 1 and TIME to zero is the the simplest use, forcing the forwarding of every received character to the user process. Visual editors use this in combination with other control bits to ensure that they see every keystroke that the user makes.

Changing the settings

It's easy to write programs that change the basic settings on the interface. Control of the terminal is one of the few pieces of UNIX state that is external to the process. If your program changes the settings, then it must be public spirited and reset them when it is done. Also, you should catch signals like Ctrl-C so that you can reset the terminal on forced exit.

Be careful when you start messing with the terminal, you can easily get the interface into a state where you cannot talk to the machine if your program dies unexpectedly. You may not be able to log off to repair things. Sometimes you must go to another terminal and issue `kill` commands to force your shell to die (use `kill -9`).

Turning echo off goes something like:

```
struct termios t_o, t_n;
..
..
/* keep old values for */
/* reset in signal handler */
   ioctl(1, TCGETA, &t_o);
   t_n = t_o;
   t_n.c_lflag &= ~ECHO;
   ioctl(1, TCSETA, &t_n);
```

Echo is now off and if your program crashes you will have to type

```
$ stty echo
```

to reset things.

Setting input raw mode is no harder:

```
/* keep old values for */
/* reset in signal handler */
   ioctl(1, TCGETA, &t_o);
   t_n = t_o;
/* leave top bit alone */
   t_n.c_iflag &= ~ISTRIP;
/* ensure 8 bit processing */
   t_n.c_cflag |= CS8;
/* turn off signal handling */
   t_n.c_lflag &= ~ISIG;
/* turn off canonical mode */
  t_n.c_lflag &= ~ICANON;
/* turn off echo */
  t_n.c_lflag &= ~ECHO;
/* forward on every char */
  t_n.c_cc[VTIME] = 0;
  t_n.c_cc[VMIN] = 1;
/* go for it */
   ioctl(1, TCSETA, &t_n);
```

System Science

Unix

SCO Unix Op. Sys (new ver)**SCO** Dev Sys.
SCO Open Desktop **SCO** ODT Dev Sys
SCO TCP/IP & NFS **FTP** PC/TCP
Interactive Unix Op Sy / Architect Series
Informix 4GL & SQL **RM** Cobol-85
LPI Compilers **C Scape** Screen Lib
LPI C++ (native) 386 Unix & Sun- **NEW**
Word Perfect for Unix **Norton Utilities**
... many more for 386 Unix and Sun

Mathematics

Derive - The Mathematical Assistant
Derive also for HP95LX Palmtop
Mathematica -Student Edition - NEW!!!
Mathematica Win, 386, Mac, Workstations
MathCad for Windows GrafTool
What's Best! **Lindo**
SPSS/PC, Statgraphics Chiwriter Sci WP

Windows Development

(see Microsoft , Borland, Zortech, Watcom)

ProtoView	£315.00
ProtoGen	£45.00
Smalltalk/V Windows	£295.00
Btrieve for Windows	£345.00
Object Trieve	£call
Win++ by Blaise	£150.00
Windows Control Pallette - Blaise	£95.00

CrossDevelopment

2500AD Cross Assemblers	£129.00
2500AD Simulator/Disassemblers	£115.00
Introl C or Mod-2 Cross Comp-6809, 68ketc	
Aztec C68K, Hitech C, Avocet	£call

Pascal (see Borland)

TopSpeed Pascal SE	£115.00
Object Professional	£110.00
Btree Filer MU	£120.00
Asynch Professional	£89.00
Blaise Turbo Vision Toolkit	£89.00
Blaise Turbo Asynch Plus	£115.00
PCX Programmer's Toolkit	£155.00

Power Basic	£89.00
many Basic libraries for comms, graphics	
PC Logo	£50.00
LMI UR/Forth	£295.00
Smalltalk V/DOS	£69.00
muLISP 90	£225.00
RM Cobol-85 (new Dev Pack)	£call.
MS DOS 5.0,	£call.
QEMM 386	£65.00

- **Prices are exclusive of VAT.**
- **Prices include delivery to GB.**
- **Prices are subject to change.**
- **VISA, Access and Mastercard welcome with telephone orders.**

Microsoft

MS C/C++ 7.0 **NEW!...NEW!**	£210.00
Quick C for Win **NEW!**	£99.00
MS Fortran 5.1	£215.00
MS Windows SDK	£210.00
MS Basic Compiler 7.1	£220.00
MS Cobol 4.5 (DOS & OS/2)	£415.00
MS Macro Assembler 6.0	£79.00
MS Visual Basic	£99.00
MS Quick C 2.5	£60.00
MS Quick C & Assembler	£99.00
MS Quick Basic	£60.00

C/C++ 7.0 competive upgrade £99.00
call for details

Authorised Languages Dealer

Borland

Borland C++ & AFW **new 3.0**	£289.00
Borland C++ (Dos / Win) **3.0**	£195.00
Turbo C++ Windows **NEW!**	£85.00
Turbo Pascal Windows	£99.00
Turbo C++ & Turbo Vision **NEW!**	£75.00
Turbo Pascal 6	£75.00
Turbo Pascal 6 Prof	£135.00
Turbo Debugger and Tools	£89.00

Authorised Languages Dealer

Nu-Mega Tech.

Bounds Checker (386)	£169.00
Soft-Ice/W (Windows)	£259.00
Soft-Ice (for 386 DOS)	£259.00
Magic CV 3.0	£135.00

Intersolv *(Authorised)*

PVCS Version Control New ver	£345.00
PVCS Config Build (PolyMake)	£145.00
PVCS Networks, OS/2 & Sun	£call
PVCS Professional Editor	£155.00

MKS

MKS Toolkit new ver 3.2	£155.00
MKS RCS & Make	£145.00
MKS Lex & Yacc**new ver 3.1**	£175.00

Lifeboat

Dan Bricklin's Demo II	£165.00
Plink 86 +	£245.00
TimeSclicer	£195.00

Tools and Editors

Brief 3.1	£195.00
Kedit (Xedit for PC) new ver 5	£115.00
.RTLink Plus	£295.00
Personal Rexx new ver 3	£110.00
Ghost (auto software testing)	£135.00
Select Case - Yourdon or SSADM	£495.00
EasyCase Plus	£315.00
CDOC	£145.00

C++ & C Compilers

also Microsoft & Borland

Zortech C++ Dev Ed (incl OS/2)	£325.00
Zortech C++ Sci & Eng	£495.00
JPI TopSpeed C++PExtend	£175.00
JPI TopSpeed C++ SE	£115.00
JPI TopSpeed C++ PE	£175.00
JPI TopSpeed C++ Pro Dev.	£229.00
Watcom C 8.5	£275.00
Watcom C386 9.0 new	£545.00
Aztec C 86 (with ROM support)	£265.00
TopSpeed C PE	£175.00

C++ Libs

Zinc Inteface Win & Dos	£195.00
C++/Views for Windows (new ver)	£call

C Datafile

CodeBase 4.5 for C, C++, Win- **new**	£215.00
Btrieve - DOS or Windows ver	£345.00
Ctree Plus from Faircom	£345.00
Faicom Professional Toolbox	£745.00

C Communications

Essential Comms	£175.00
Greenleaf CommLib	£195.00
C Asynch Manager (Blaise)	£115.00

C Screens

CScape with Look & Feel	£365.00
Vermont Views 3	£335.00
Panel Plus II	£275.00

C Graphics

Essential Graphics & Chart	£195.00
PCX Programmers Toolkit	£150.00
GX Graphics	£125.00
Halo	£195.00
Metawindows	£195.00

Misc. Libs

C Tools Plus	£89.00
Bar Code Library- source	£245.00

... many more libraries

Pharlap **386** DOS Ext with **ASM**	£325.00
Pharlap **286** DOS Ext SDK	£325.00
Pharlap 386 ASM/Link/LOC	£775.00

Fortran Compilers

Lahey F77L	£375.00
Lahey F77L-EM/32 with OS386	£875.00
Watcom F77 386 8.5 with ADS supp	£495.00
Salford FTN 77-386	£765.00
PC/Interacter (screens)	£325.00
Ingraf Graphics source	£195.00

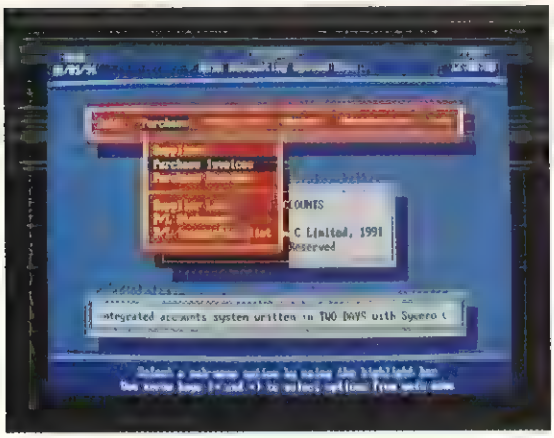
... many Fortran Libraries

3-5 Cynthia St
London N1-9JF
Fax: 071 837 6411

071 833 1022

• Software • Specialists • Software • Specialist • Software • Specialists •

SoftwareDevelopers!



...Here's the same application half an hour later usingTurbo C ++'s advanced VROOM technology...

...and here's the same application an hour later running under SCO UNIX, with no runtime licences required !

Produce a sophisticated application like this in Microsoft C , with either BTRIEVE or C-ISAM, running under DOS in just a couple of days!

...Another twenty minutes and it can access dBASE files...

This entire application was developed in a couple of days using **Sycero C**. Sycero is a powerful program generator that **dramatically** reduces your programming time. Sycero C generates fully structured and documented C source code to be compiled with Microsoft or Turbo C. Sycero C supports the Btrieve/C-ISAM file handling systems, plus it can even generate C programs for use with dBASE or Clipper files and indexes. Sycero C UNIX allows you to develop software to run under SCO UNIX, and you can regenerate existing Sycero C DOS-based applications with absolutely **no changes** required to your definitions.

For more information, either call us on **0622 691616**, or fill out and return the coupon.

CIRCLE NO. 957

dBASE is a trademark of Ashton-Tate All other trademarks acknowledged.

Please send me further details about Sycero C

Name: .....

Company: .....

Address: .....

.....

.....

.....

Tel: ..... Fax: ..... EXE 5/92

I am also interested in your generator for Clipper, Sycero dB ☐

System C Ltd 60-61 High Street Maidstone Kent ME14 1SR

Tel 0622 691616 Fax 0622 691241

If you do all this and your program crashes then you might be able to reset things by typing:

```
% stty sane
```

Finally on this topic, beware that some shells reset the terminal interface when they get hold of it. However, if you do alter terminal settings you must never rely on the shell resetting things after you. Some shells will not.

Writing

This is much less complicated. A user process will use the `write` system call to move characters from its address space into the kernel. The system call will return when all the characters have been transferred. It does not wait until all the characters have been output.

The process will block if the kernel has insufficient resources to take the data.

The only complication with output is the support of per character delays so some types of hardware can function correctly. The `c_oflag` word in the `termios` structure controls character delays. You rarely touch this in programs. It tends to be established by the user at login time.

Multi-processing

On a multi-processing system, we do have to worry a little about what happens when more than one process attempts to write or read from the same terminal.

Several user processes can be using the top level in parallel. They will be executing the same code but running on their own stacks, so each process is running a separate instance of the top level routines. All these data sources will be adding characters to the data buffer that interfaces between the top and bottom levels. The bottom level ships the characters to the output device. The effect is a jumble of output from several different sources.

Some systems help by allowing the user to specify that background processes will be blocked if they attempt to talk to the terminal. This is a 'BSD-ism' and was added as part of job control.

It's easy to set up two processes reading from the terminal. This has little semantic value and the effect is hard to predict. In cooked mode, each process is likely to get alternate lines from the user. This can result in some confusion so it's a good idea to avoid this.

Further reading

See the relevant manual pages on your system: `stty(1)` and `termio(4)` are good sources. Poke about the include files on `/usr/include`, look at `termios.h` and `termio.h`. For more discursive material see a System III or System V book: I am using *UNIX System Programming* by Keith Haviland and Ben Salama, published by Addison Wesley. The POSIX standard P1003.1:1990, now published as ISO/IEC 9945-1 is also a source on this material.

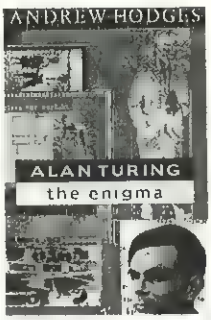
EXE

Peter Collinson is a freelance consultant specialising in UNIX. He can be reached electronically as `pc@hillside.co.uk` *(although your mailer might be happier to put the address the other way round) or by phone on 0227 761824.*

Books

This month, an autobiography of a pioneer.

Of machines, questions and breasts



'Turing' is a name which conjures up a vague sensation of awe and respect, yet I believe many of our profession know little about the man, his life and his work. The new edition of Andrew Hodges brilliant biography *Alan Turing: The Enigma* gives me a chance to recommend a book which is not only an excellent read, but also gives one an unusual perspective on the business of writing programs.

Turing's story is a fascinating one. Born in 1912 to a middle-class 'empire-building' family (his father was in the Indian Civil Service), most of Alan Turing's early years were spent in England with friends of the family and at various boarding schools, culminating with Sherborne, a league division two public school situated in East Dorset. Here he suffered considerably under a regime that despised maths and 'stinks' and individualism, and favoured 'footer'-playing conformists. Despite early mathematical efforts, which included the feat of finding a series for $tan\text{-}1\ x$ without having learnt any calculus, Turing was nearly expelled from Sherborne midway through his career for poor work in English and Latin. In the event he was saved only by a chance change in headmaster.

Notwithstanding these struggles, Turing obtained a scholarship to King's College, Cambridge in 1931. At Cambridge he prospered; his proof of the Central Limit Theorem (yes you do - it's the one which allows the Normal curve to be applied to just about any random variable) was rewarded by an election to Fellowship in early 1935. In fact, the result had already been proved in 1922 by another mathematician - Turing had absent-mindedly neglected to check to see if the work had already been done.

It was now that Turing did his most famous mathematical work. In 1928 the great mathematician David ('matrix') Hilbert had posed a number of questions. The third of these was: was mathematics 'decidable'?, by which he meant 'did there exist a definite method which could, in principle, be applied to any assertion, and which was guaranteed to produce a correct decision as to whether the assertion was true?' Turing applied himself to this question and, in order to help in the abstraction of the problem, conceived his famous Turing machine.

The Turing machine is well-described elsewhere; essentially it is an imaginary device which can manipulate and obey symbols written on an infinite length of paper tape. The similarity to a CPU operating on a (sadly finite) length of address space is striking. At this stage, however, Turing was only using his design as a 'prop' to help answer a beautifully esoteric mathematical question. Opponents of 'blue sky research' on the basis that it doesn't help with practical problems should take note.

The war came, and Turing was seconded to Bletchley Park, Buckinghamshire, to join the Government team devoted to cracking the Germans' military cyphers, and in particular breaking an encoding device known as the 'Enigma machine'. Turing and his team acquitted themselves brilliantly, and by May 1941 was able to provide the Admiralty with decyphered traffic from the predatory U-Boats in the Atlantic, and thus protect Britain's vital line of supply to the US. Such was the importance of Bletchley Park that in October 1941, when Turing wrote a letter appealing for extra staff to Winston Churchill himself, he secured his needs at once on the Prime Minister's orders.

Towards the end of the war, Turing became somewhat remote from the day-to-day business of code cracking, and began to pursue thoughts of his own. The knowledge and experience of electronics he had gained decyphering codes were combined with his idea of an abstract 'Turing machine' - he invented the 'electronic brain'. When peace came, he joined the National Physical Laboratory, and wrote a report, 'Proposed Electronic Calculator', which described his machine - a general-purpose digital computer.

Given that the report was written before any computer existed, the depth of his vision is startling. Here is his description of the call/return/stack mechanism for subroutines: 'When we wish to start on a subsidiary operation we need only make note of where we left off the major operation and then apply the first instruction of the subsidiary. When the subsidiary is over we look up the note and continue with the major operation... [One way to do this is] to keep a list of these notes... with the most recent last.'

In some sense, after the completion of this report, nothing ever went quite right for Turing again. The NPL dragged its heels over the building of his machine, with the result that Manchester University was able to produce the first computer before the first valve had been plugged into his own machine. He moved to Manchester and contributed some early programs, but his heart wasn't in the work. His approach to programming was brutally primitive - he worked in machine code written as base 32 numbers written backwards, and never appeared interested in high-level computer languages.

In the end, Turing's homosexuality brought catastrophe. He foolishly reported a lover to the police for attempted theft. His secret was discovered, and he was charged and convicted of gross indecency. The barbaric punishment took the form of treatment with female hormones; in order to be effective in the 'reduction of libido', it was believed that the dosage had to be sufficient to cause the subject to grow breasts. Although he saw out this treatment, he seems to have failed to have re-established his life, and two years later he committed suicide.

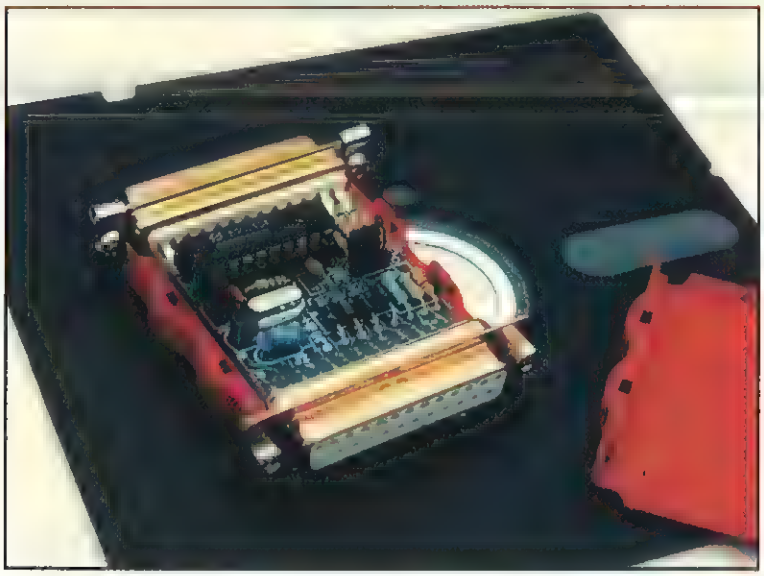
Hodges brilliantly captures Turing's lonely figure; his descriptions of the technical issues are neither too light nor too complex; and, damn it, it's a dead good story. Definitely worth acquiring for the annual trip to Majorca.

Title: Alan Turing: The Enigma — *Pages: 586*
Publisher: Vintage — *Author: Andrew Hodges*
Price: £8.99 — *ISBN: 0-09-911641-3*

Books Received This Month

ANSI and ISO Standard C	P Plauger and J Brooke	Microsoft Press	£14.95	ISBN: 1-55615-359-7	278pp
TEX for the beginner	Wynter Snow	Addison-Wesley	£22.95	ISBN: 0-201-547-996	377pp
Object Oriented Programming in C++ and OSF Motif	Douglas Young	Prentice Hall	£24.95	ISBN: 13-630252-1	434pp

Announcing DeadLock III



Don't think Dongle, Think DeadLock

- Unbreakable

Also available - DeadLock I
Prices from £8.75 to £16.50

Technical

- ★ Sits on parallel port
- ★ Completely non-intrusive and transparent
- ★ Programmable
- ★ The hardware component is based on a custom-designed IC
- ★ There's full DeadLock support for today's most popular languages, compilers and development environments
- ★ DeadLock contains all the logic necessary for password control and to perform calculations
- ★ DeadLock comes with a special PROTECT utility that allows you to build an 'envelope' of protection around your product's COM or EXE files
- ★ Each DeadLock key can be programmed by you, to transform it into a protection device that's unique to your product
- ★ Easy to use manual
- ★ May be used for joint ventures
- ★ DeadLock comes with a variety of friendly, menu driven utilities
- ★ Different coloured cases available to suit your preference

Distributors required worldwide. Please send CV to address below.

Please address all enquiries to:

BL Computer Security Limited
101 Hendon Lane, Finchley, London N3 3SH
DeadLock
PO Box 2543, London N3 3UA, United Kingdom
Fax: 081 346 2672 Tel: 081 343 0734

CIRCLE NO. 957

NEW DESKPRO 386e
M40, 16 MHz, 40Mb HD, 2Mb Memory.
£995 + VAT

THRIPPLEWOODS

FAMOUS FOR VALUE IN MICRO COMPUTERS

OPENING TIMES
MON - FRI 9.00 - 5.30
SAT 9.30 - 2.00
24HR ANSWERPHONE

THE END OF RANGE & EX DEMO SPECIALIST

AMSTRAD ITEMS SUPPLIED

AMSTRAD PRICE LIST

	R.R.P. £	THRIPP £
FAX MACHINES		
FX 5050 Fax Machine		
On Site Warranty	349 00	**319.00**
FX 6000 AT Fax Machine		
On Site Warranty	425 00	**389.00**
WORD PROCESSORS		
PCW 9256 Word Proc.	349.00	**319.00**
PCW 9512 + Daisywheel + ASF	449.00	**409.00**
PCW 9512 + Canon BJ10e	549 00	**509.00**
LAPTOPS		
ALT 286/40	799 00	**699.00**
ALT 286/40 Laptop + BJ10e	999 00	**879.00**
ALT 386 SX/80+2 Laptop	1399.00	**1149.00**
ACL 386 SX/120+4 Colour L/top	3999.00	**3279.00**
BJ10e Bubble Jet Option	200.00	**184.00**
NOTEBOOKS		
ANB 386 SX/20 Notebook	1199 00	**989.00**
ANB 386 SX/40 Notebook	1499.00	**1199.00**

12 months On-Site Warranty
PCW & Desktop PC's £25

	R.R.P. £	THRIPP £
1512/1640 ITEMS		
Amstrad 4 Games PC 1512/1640		**4.95**
PC1640 Technical Ref Manual		**5.95**
LINKMASTER for PPC512/640 to PC Comms		**5.95**
3000 SERIES DESKTOPS		
PC 3286 SD30 3 5" FDD with Samsung 105 MB HD		**549.00**
PC 3286SD 5 25" FDD		**309.00**
PC 3286SD 5.25" FDD Samsung 105Mb HD		**499.00**
PC 3286 HD40 3.5" with Lotus1-2-3 v2.2	450.00	**399.00**
PC 3286 HD40 3 5" & 5.25" FDD's with Lotus 1-2-3 v2.2	500 00	**449.00**
PC 3386 SX HD40 with MS Windows & Mouse	650.00	**589.00**
PC 3386 SX HD80 +4 with MS Windows & Mouse	900 00	**799.00**

HARD DISKS ON CARD Great for 1512/1640	
20Mb Western Digital	**99.00**
30Mb ATASI	**129.00**
96Mb Seagate	**249.00**

	R.R.P. £	THRIPP £
AMSTRAD NETWORK LAN PACK		
Connect 3 PC's for		**99.00**
D BASE II for PCW		**9.95**
5000 SERIES DESKTOPS		
PC 5086 HD40 3.5" with Microsoft Works	300.00	**269.00**
PC 5286 SD with Lotus 1-2-3 v2 2	350 00	**299.00**
PC 5286 HD40 3.5" with Lotus 1-2-3 v2 2	450 00	**399.00**
PC 5286 HD 105 3.5"		**499.00**
PC 5286 Family Pack with Microsoft Works 2	680 00	**619.00**
4000 SERIES DESKTOP		
PC 4386 SX HD80 +4	900.00	**799.00**
MONITORS FOR 3000, 4000 & 5000 SERIES		
10" Mono VGA Monitor	99 00	**89.00**
14" Mono Flat Screen VGA	99.00	**89.00**
14" Colour VGA Monitor	199.00	**179.00**
10" Colour Super VGA	299.00	**269.00**
14" Colour Super VGA	299.00	**269.00**
12" MonoVGA	99.00	**89.00**
Lockable 5 25" Disk Box		**9.95**

NEW 5286 WITH 105MB HD

Features 5286 3.5" FD with a Samsung 105 mb 16 ms Hard Disk, Perfect for applications where you need to store a lot of data.
RRP £549 **OUR PRICE £499**

MAINLAN EASY Networking Starter Kit. Contains all the hardware, software + cabling you need to connect two PC's 199.00
Easy Extension Kit. Add another PC to you network for only 99.00

WINDOWS 3.1 UPGRADE.
A Kit to allow V3.0 users to U/G to V3.1 for only 45.00
WINDOWS 3.1 FULL PRODUCT with FREE Adobe Type Manager + 4 fonts 99.00

OTHER ITEMS

COMPUTERS

IBM COMPUTERS
PS/1 101 80286 512K 1 3.5" Disk, VGA Mono Monitor Free Microsoft Works 2.0 **449.00**
PS/1 201 80286 512K 1 3.5" Disk, VGA Colour Monitor. Free M/S Works 2.0 **549.00**

OPUS COMPUTERS
PCSX, 80386, 16MHZ. 40Mb, 3.5' Drive, 2Mb RAM with VGA Colour Monitor. 2 Serial & 1 Parallel Port, 8 Expansion Slots. 1 year Free on-site maintainance. **849.00**
As above with 135Mb Hard Disk **999.00**
PCIV, 40Mb, 80286, 40Mb, 3.5' Drive, 2Mb RAM with VGA Colour Monitor. 2 Serial & 1 Parallel Port, 8 Expansion Slots. 1 year Free on-site maintainance. **649.00**

NEW DESKPRO 386e M40 16 MHz, 40Mb HD, 2Mb Memory **995.00**
POQUET PC Palmtop Computer **399.00**
SANYO 286 60Mb Notebook **799.00**
TOSHIBA T1000 LX Notebook **399.00**
TOSHIBA T1600 /40Mb 80C286/ 12MHz, EGA Portable **399.00** BOTH EX-RENTAL

AST Bravo 286 Model 45 8mz 80286 processor. 4 ISA 16 bit slots. 1 serial and 1 parallel port. 640K RAM upgradable to 16Mb. 40mb Hard Disk. **£399**

Portable 386 40mb external tape unit. **£499**

Compaq Portable 386 Model 110 (5.25" disk). Dual mode plasma display 20mz 80386 single 5.25" disk100mb fixed disk. NEW BOXED **£2999**

Compaq Deskpro 286e Model 20 (5.25" Disk)1mb RAM, 20mb fixed disk 5 expan-sion slots.Integrated VGA controller serial, parallel and mouse ports. NEW BOXED **£699**

Compaq Deskpro 286e Model 40 as model 20 but with 40mb hard drive. NEW BOXED **£799**

Toshiba T3100e EX RENTAL **£799**

Toshiba T3100sx EX RENTAL **£1199**

Toshiba T5200 Model 40 EX RENTAL **£1299**

Poquet Palmtop PC. THE Poquet PC - 640K CGA LCD Screen. 100 hours use from 2 AA batteries NEW BOXED £399
External 3 5" floppy **£159** PQ Link Cable **£19**

IBM PS/2 Model 50-Z Computer. 10mz 80286 with 1 mb RAM. One 1.44 Mb 3.5" disk, to which can be added a second drive.
One 60mb Hard Disk 3 Expansion slots VGA Standard with serial, parallel and mouse ports. Uses 85ns memory giving 25% speed improvement over 50-021. Requires Keyboard and Monitor. **£799**

IBM PS/2 Model 70 386 - 121.20mz 803K86 processor 2mb RAM upgradable to 6mb.One 1.44 disk to which can be added a second One 120mb Hard Disk 23ms access time 3 expansion slots VGA standard with serial,parallel and mouse port Requires Keyboard and Monitor. **£2299**

IBM PS/2 Model 80-M21, 80386, 20Mhz with 320Mb HD EX DEMO **£1799**

Sanyo 386SX. 60MbNotebook EX DEMO **£999**

ADD-ONS

EPSON VGA GRAPHICS CARDS 69.00
Suit most PC's

ACCESSORIES

B+R MAINS FILTER 13 amps. Don't put up with fluctuating mains current use one of these 13 amp plugs to smooth the power to your PC and help avoid data damage **5.95**

THUNDERBOARD SOUND CARD
Compatible with Sound Blaster + Adlib, 11 voice FM music synthesiser, Built in Joystick Port and 2 watt amplifier. **89.00**

CIPHER 60Mb Back-up systems **Ex-demo 249.00**

PRINTERS

Epson FX 100+	EX RENTAL	**£99**
Epson FX 105	EX RENTAL	**£119**
Epson FX 1050	EX RENTAL	**£299**
Epson FX 1000	EX RENTAL	**£199**
Epson FX 2500+	EX RENTAL	**£299**
Epson FX 2500	EX RENTAL	**£349**
IBM Proprinter XL	EX RENTAL	**£149**
IBM Quietwriter III	EX RENTAL	**£399**
Kodak Diconix 150	EX RENTAL	**£149**
NEC Pinwriter P7+	EX RENTAL	**£299**
OKI Microline 390	EX RENTAL	**£199**
Qume Sprint 11/55+	EX RENTAL	**£199**
Epson DFX 8000 Amazing 1066 cps!	EX DEMO	**£1499**
Paintjet Serial	EX DEMO	**£379**
7475 Plotter A4/e A3/B	EX DEMO	**£499**
IBM 4019-001 Laser Printers. 10 page per minute	EX DEMO	**£599**
IBM ExecJet 4072	EX DEMO	**£459**

BACK-UP UNITS & TAPES

Sysgen Image tapes **£9**
Irwin External 40mb 400Z tape drive **£149**
Irwin 120 Tape backup for PC/XT **£169**
Irwin External 20 Mb Backup for XT/AT286. Requires interface board. Drive **£79** Interface **£79**
Irwin Internal 20 Mb Backup PS/2 Requires fitting Kit. Drive **£79** Fitting Kit **£66**
Bernoulli 8" 20mb Tripack. Pack of 3 20mb removable cartridges **£179**

COMPUTER EXPANSION

Auric Busman 5500 for Sharp PC 5541 **Auric Busman** 5500 for Sharp PC4602/4641
Each offers 2 addition expansion slots NEW BOXED **£249** each

Compaq 300Mb external Disk Drive unit. Comes with 300Mb Disk Drive, and can hold an additional 300Mb Drive. Compatible with all Deskpro Models **£1799**

Compaq Equipment
5.25" in 1.25mb drives. For 386s/20c and also for 386/33, 486/25 and systempro EX DEMO **£99**
NEW SLT Replacement Keyboards **£69**
NEW US SLT Keyboards **£69**
LTE/386S 1mb memory exp EX DEMO **£99**
SLT 1mb Memory expansion EX DEMO **£69**
NEW Deskpro enhanced keyboards **£129**
NEW Asynch comms/clock board for OLD Deskpro **£39**
Portable 386 1 2mb D/sk D/vs EX DEMO **£79**
Nylon Covers for386/20e with Keyboard and Monitor **£9**
Advanced ESDI controller for adding 320 or 650mb EX DEMO **£249**
Portable 386 1 to 2mb expansion board EX DEMO **£199**
5 slot Expansion chassis for older Toshiba Notebooks except T1000 and T3200 EX DEMO **£299**

DISK DRIVES

IBM Optical Disk drive 3363 B01 For PC or PS/2 as a second or subsequent Disk Drive **£599**
Toshiba External 360Mb 5.25" FDD EX RENTAL **£39**
NEW 40Mb Hard Disk for IBM 55SX **£199**
60Mb PS/2 Hard Disk 27ms access time EX DEMO **£299**

EXPANSION BOARDS

Profit Elite 16mb Expansion board for AT and 386 machines. NEW BOXED **£699**
Profit Elite 16 Hypercache. 32K high speed memory cache. True 16 bit memory access in

CIRCLE NO. 958

all EMS regions. Expands to 16mb using 1mb or 256K SIMMS using 120ns chips 2 serial and 1 Parallel port. Also supports OS/2 XENIX/ UNIX and LANS.
NEW BOXED **£149**

Profit Elite 16/2 (512k to 16mb) PS (MCA) IBM Models 50 to 80. Delivers up to 16mb of OS/2 or EMS 4.0 Memory in a single slot.
Utilises both 256k and 1mb SIMM memory
16 * 256k modules gives 4mb
16 * 1mb modules gives 16mb
Mix and match both module types Ability to utilise multitasking in OS/2 and Windows
Unram software for switchless installation
NEW BOXED **£99**

FAX / MODEMS / MULTIOPLEXER

LINNET 1200 PC MODEM Basic V22 1200 baud full duplex. **Ex-demo 9.00**

Multimodem External 9600bps Modem. Full DuplexCallback security. Help menus. Non volatile memory for up to 10 'Phone numbers. NEW BOXED **£229**

Quasercard 2400 BPS Internal modem for **IBM PS/2** MMP error correcting modem for **PS/2** with level 4 error correction. **£199**

Breakout 2400/2 PS/2 Internal Modem. A Quasercard as above plus PCL's award winning software and a copy of Crosstalk XVI

EPSON AX-2400 V42 Modem for Portable **£199**

Unity Gold V22bis half length modem Supports MNP level 5 EX DEMO **£149**

EPSON VGA Adaptor. **£69**

GRAPHICS CARDS

VGA CARDS ATI VGA Wonder/265K 16 Bit VGA Graphics Card 1024 x 768 up to 256 Colours 132 Column text Mouse Port included. Drivers include Autocad, Windows, Ventura, Lotus and Gem. A 'Branded' product at an excellent price! **£99**

LANS

Token Ring PC Adaptor II. **£299**

Micronyx Triumph! Workstation Manager (Token Less). Security software for networks. Includes Administrators guide and Users Guide. **£89**

Micronyx Triumph! Workstation Manager (with Token). Comes with Token to provide maximum data security by encription. Includes Token and Token receptacle and cable. **£99**

Browser Audit Analysis PC. Used with Micronyx Triumph! or Trispan to provide information about systems use, program utilisation, attempted system entries etc Gives full audit trial of all system transactions. **£69**

Ethernet controller cards *bit PC controller with Novell Netware drivers **£149**

Ethernet M.P.S. Controller. Multi protocol support. **£169**

N series Transceiver. Allows connecting to IEEE 802 3 can. **£99**

Omninet PC/NOS Software V1.1. Supports only Corvus Omninet Transporter cards. Allows sharing of hard disks, printers etc **£49**

LASERJET PERIPHERALS

CPI Jetmemory for HP Laserjets.Functionally compatible with HP memory cards 1mb and 2mb starter II and IID. Plugs into LaserJjet II or IID Recommended for single users with small amounts of font and graphics use 1mb **£69** 2mb **£99**

Plus 1mb Expansion II and IID.Combines with Jetmemory 1mb starter for 2 5mb total memory. Recommended for single users with large font and Graphic usage. **£49**

Plus 3mb expansion II and II.Combines with Jetmemory 1mb starter for 4.5mb Total memory. Recommended for multiuser environments with large combinations of fonts and graphics. **£129**

Plus 2mb expansion II and IID. Combines with Jetmemory 2mb starter for 4.5mb total memory. Recommended for Multiuser environments with large combinations of fonts and graphics. **£99**

FONTS for II and IIDJet font Superset II and IID 216 Equivalent HP fonts. **£49**

Jet font "H" Courier 10 Legal Font. **£39**

Jet Font "R" Presentation and Lettter Gothic. **£39**

Jet Font "T" Helvetica Port. **£39**

Jet Font "U" Helvetica Port Roman. **£39**

Jet Font "V". **£39**

Laserjet 1mb memory for IIP and III. 1mb memory board for Laserjet IIP and III. **£59**

Jetpage II.Postscript emulation in a cartridge. Provides 35 scaleable font families equiv. to Apple LaserwriterNT. Requires 1mb memory expansion. **£149**

Jetpage IID, IIP, III and IIIDAs Jetpage II. **£149**

Canon Laserjet Elite PRESTIGE Font Cartridge **£129**

HCF 1000 sheet feed for Laserjet II and IID and equivalents. Inc Canon and Brother **£149**

Laserjet Plus/500 2mb Upgrade. Gives dual ports serial and Parallel and 2mb additional memory **£499**

IBM 4019 10 page per minute upgrade **£499**

IBM Model 115 and 125 8mb Fast memory card **£999**

For use with IBM personal page printer adaptor to support an attached 4216-020 printer Provides Postscript Language interpreter and typefaces **£349**

LASERJET

HP Scanjet Sheet feeder. **£299**

HP Laserjet Proprinter/Epson Cartridge. Emulation cartridge for Laserjet IIP or IID Does not work with Laserjet II or III. **£69**

HP Laserjet EX RENTAL **£399**

HP Laserjet plus EX RENTAL **£429**

Toshiba Page Laser 8 EX RENTAL **£599**

MICE

Microsoft Bus Mouse with Windows 286, and PC Paintbrush. **£99**

PRINTERS ADD-ONS

SStar 15" single bin sheet feed for NB range printers. NEW BOXED **£19**

Epson RX80 F/T cut sheet feed. NEW BOXED **£9 95**

Epson FX 80/85 Cut Sheet Feed. NEW BOXED **£19**

GQ 5000 250 sheet cut sheet feeder NEW BOXED **£99**

Epson EX800/1000 Colour option unit. Gives up to 7 colours for presentation quality output. **£9**

PLOTTERS

IBM Plotter 6180-001 Compact easy to use high resolution desktop plotter with 8 pen carousel with availability of two types of fibre tip pens in 10 colours. Uses 8.5" x 11" paper and transparancy film. Automatic capping extends pen life. **£999**

SCANNERS

Interface cards for Scanjet Plus AT or MCA EX DEMO **£199**

Sharp JX 300 A4 Colour Scanner EX DEMO **£649**

Mac Interface for above **£299**

PC Interface for above **£299**

VIDEO MONITORS

Dimension 19" Mono Display EX RENTAL **£199**

IBM 5151 PC Display EX RENTAL **£99**

IBM PS/2 8503 Mono Display EX RENTAL **£49**

Radius Mac 19" Mono Display EX RENTAL **£199**

SuperMac 19" Mono Display EX RENTAL **£199**

IBM PS/2 8512 VGA Colour Display EX RENTAL **£129**

IBM 8515 Monitor 1024 x 768 Res EX DEMO **£399**

SOFTWARE

PASCAL SOFTWARE. Turbo Pascal V3.0 PC. This is the Easy to use Pascal compiler, at a bargain price! **£34**

Turbo Pascal Professional PC V5 5 Includes assembler and Debugger. **£99**

Turbo Pascal Developers Library PS. **£129**

Borland Word Wizard PC. Technical reference for Turbo Lightning as well as Turbo Pascal. Source code. **£39**

Sidekick PC. A handy utility which can, at the touch of a key, provide a calculator, notepad, appointmet calendar, auto dialler and ASCII table **£29**

Caere Omnipage 386 v2.12 PC/PS Lets personal computers recognise text from virtually any scanned documet regardless of the number of columns, types of font, use of bold, italic text etc. The text from almost any business document or page can then be saved in many popular spreadsheet or database formats. Requires 1mb conventional and 3mb expanded memory, plus QEMM 386 V5.11 or Qualitas 386 MAX v5.0. **£399**

Omnipage 386 V2 1 also available for **£249**

GEM Artline, from Digital Research. Powerful graphic design capability with simple to use interface. Graphics tools include Move,Copy, Rotae, Scale and Mirror Image support for .IMG, PCS and TIFF files, Large LIB of pre-generated drawings. Fully compatible with Xerox and Aldus Pagemaker. NEW BOXED **£199**

Davrell PC. A powerful graphics management system for getting your message across either on screen, in hardcopy, on OHP foils through DTP or Videoshow. Data can be directly keyed in, or taken from 123, Symphony, DIF, ASCII or PIC files. Design rolling demon-strations. Generate reports from spread-sheets, create charts, and link them to your spreadsheet. Every time the data changes, the charts change too! NEW BOXED **£199**

Corel Draw V1 2 PC Draw freehand, autotrace, import an image, add type, (100 fonts included and 4300 supported).Scale, Rotate, Mirror, Fit text to a curve. Stunning 3D and Postscript effects. Full Pantone Colour. Includes Videotape Tutor. NEW BOXED **£249**

Autoroute Lan server PC/PS. Provides the basis for putting Autoroute on a LAN. Allows one user. Additional 5 User packs can be bought. NEW BOXED **£299**

Funk Noteworthy PC/PS A cell annotation program that works with Lotus 123 and Symphony Allows you to create pop-up notes for individual cells on a worksheet and call them up with a single keystroke. Import feature to incorporate text files. Automatic load and save with spreadsheet. **£39**

IBM Displaywrite 4/2 PC IBM's word-processing Program! **£199**

Migent Page Ability PC. DTP Package-WYSIWIG. Line and Object drawing tools. Direct input of scanned images. Unlimited number of documents. Unlimited number of typefaces per page. Font sizes from 6 to 30 supplied, 22 fonts in 3 typefaces. **£49**

Migent ability intgrated software for PC Word Processing, Database, Spreadsheet **£9.99**

Microsoft Windows 286 PC. **£39**

Microsoft Cobol Compiler. ANSI 85 Standard Code compatible with Xenix 286 5 25" disks. **£299**

Microsoft Word V4.0 10 User pack includes One full product and 9 sets of 5.25" and 9 sets of 3.5" disks. **£699**

Microsoft Excel 10 User pack. Includes 1 full product and 9 additional sets of 5.25" and 3.5" disks **£699**

Microsoft Word for Windows V1.0 PC/PS **£99**

Microsoft Fortran Compiler V4.1PC Optimising Compiler with Library manager Full ANSI Standard with DEC and IBM extensions. **£149**

Central Point Back up V6 **£9**

WinConnect Windows Desktop to Desktop or Laptop comms package. Enables Laptop to work as a server to Desktop. With this software you can turn an old PC into a dedicated server to your laptop! EX DEMO **£49**

Printer Boss An advanced printer utility for dot matrix printers. Letter quality printing, sideways printing, custom font design, alternative character sets, printer format control, RAM printer buffer up to 32K, typewriter output mode, printer string set up. **39.00**

PFS FIRST PUBLISHING V3.0
Has all the features needed to make top quality publications easily. Hi-res fonts included. Requires only 512K, but 640K recommended. **129.00**

LOTUS 1-2-3 V2.01 3.5" **49.00**

LOTUS 1-2-3 V2.01 5.25" **49.00**

SUPERCALC 4 LAN **19.95**
Local area network version of SUPERCALC 4 also works on stand alone PCs

Digital Research GEM Artline Produces designer quality artline. Large Library of pre-generated drawings. Fully compatible with DTP systems. **199.00**

PC TOOLS V6.0 with voucher to allow upgrade to V7.0 for £20 **29.00**

Alpha Four V1.1 Database Designed to be the most powerful database available for non-programmers. Link-up to 10 related datbases. Application generator. Compatible with dBase. Mail Merge feature. **Ex-demo 39.00**

Central Point Back-Up PC **9.00**

R+R Relational Report Writer for dBase III or any .dbf files. Prepares reports faster, easier, without programming. Sales, most active accounts, inactive inventory purchase orders, multi page invoices etc. **39.00**

MULTIMATE II 1.0D (1990)Word Processor. Famous Multimate includes: Pull Down Menus. 40,000 Word Thesaurus with 470,000 synonyms. Comes with English Spell Checker. Six maths functions including percent. Direct merge with .dbf files **49.00**

SWITCH BOXES

IDL 4 ports to 2 Ports parallel switch.Used to connect 4 parallel devices to 2 parallel devices eg 4 computers to 2 printers.Detects data from any of the 4 ports, but printer desired is manually selected. **£89**

IDL 2 ports to 1 port Serial switch. **£49**

Verran A/C Datalink. **£99**

3270

Keytronics 3270 Keyboard with UK Keycaps For all 3270 users, your own 3270 Keyboard! **£199**

PCOX/Multi PS. Provides terminal emulation software for up to five 3270. Keyboard/ display and printer sessions. Emulates 3278/79, 3270 PC, 3270 PCG and 3287 Printer. Requires Coax adaptor for PC or PS/2. NEW BOXED **£399**

Tel: 0706 832005 or 071 265 4565 Fax: 0706 832100

All Prices exclude VAT and Delivery. All Prices subject to change without notice. Access/ Visa/ Mastercard accepted. Official orders accepted.

Unit 4, Unicom Park, Carrs Industrial Estate, Haslingden, Rossendale, Lancs. BB4 5HU Tel: 0706 832005 Fax: 0706 832100

We Have a Right to Know

Francis Glassborow discusses the practice of keeping compiler bug-lists secret.

I am not sure what you would like me to write about in future columns so please let me know if you would like something different. For now, I will assume that you have effective ways of dealing with your coding problems and so I will concentrate elsewhere.

One of CUG(UK)'s primary objectives is to campaign for better programming and better support for programmers. I will return to both these in future columns but this time I want to concentrate on the refusal by compiler writers to give us information we need. It is disgraceful.

Those who know me will be familiar with my claim that 'debuggers' are for finding bugs in the compiler. Those who know me well know that I mean it. In any other sphere of human endeavour a manufacturer who stayed silent about a potentially fatal flaw in a product would not remain in business - the damage suits would see to that. Not so in computing. Compiler manufacturers, almost without exception (Zortech was one of those rare ones), refuse to let us know of faults found in their products. The result can vary from hours of lost development time to serious damage to a developer's reputation.

As programmers, we know that programs have bugs just as 'typos' creep into the most carefully sub-edited text. We are skilled at finding ways to work around known bugs, sometimes too skilled for our own good and the work-around falls over when the compiler is fixed. We should not have to waste time with rediscovering bugs that are already known to a product's authors.

It has got to the state where many programmers no longer report bugs because they are fed up with the frequent reply 'Oh, we already know about that one.' That being the case, why didn't they tell us? Yes, I know the answer to that and it is just about as complimentary to your professionalism as consultants who suggest you are just games players (see letters in *.EXE* April '92).

The minimum registered users of a product should expect is regular information about bugs. Fixes would be nice but I will settle for being treated as an adult who can be trusted with the information that software is not perfect.

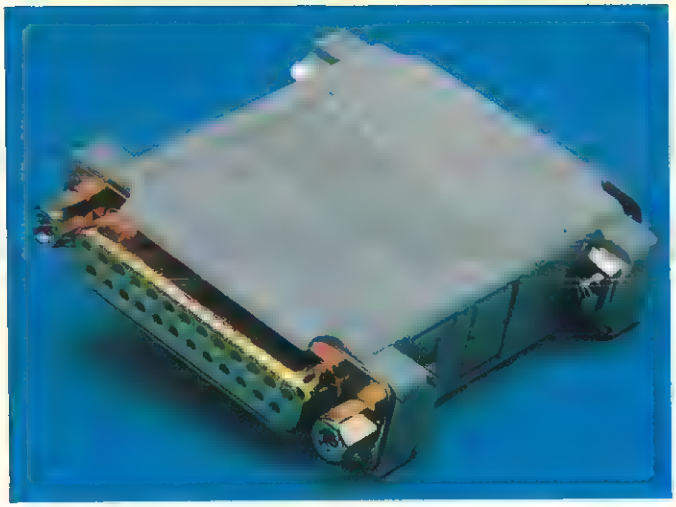
Programmers and software development managers should demand that the compiler producers acknowledge their 'Right to Know of Bugs.' That will make all our lives easier and reduce the log-jam on some technical support lines.

Now for something else. If you are coming to the 'C Plus C++ Conference' at Gatwick Hilton in June, please look me up as I shall be attending all week. The conference program looks excellent. I have enjoyed listening to many of the speakers in the past. The problem will be choosing between concurrent events.

The one event that you will have no problem with is an open meeting of CUG(UK) starting at 6 pm (just after the exhibition closes) on Thursday, 11th June.

.EXE

For information about CUG(UK) write to 64 Southfield Road, Oxford, OX4 1PA or ring 0865 246490.



How many users of your software paid for it???

SOFTLoK International Limited was established in 1987 with the introduction of our SOFTLoK and SOFTLoK PLUS devices to combat the ever increasing problem of software piracy. Our SOFTLoK range of software protection devices are used by hundreds of software developers from small consultancies to large multinationals. To cope with the ever changing needs of our customers we have developed SOFTLoK II which combines the programmable features of SOFTLoK PLUS with a low unit cost similar to our original SOFTLoK product.

SOFTLoK II™

The Next Generation

SOFTLoK II units are programmable devices containing read/write memory protected by a password. Both the memory and the password can be changed at any time using our routines in your application software. Easy to use menu-driven software is provided to allow small or large batches of SOFTLoK II units to be programmed with their initial data & passwords ready to be sent out with the protected software product.

As SOFTLoK II units plug into the parallel printer port they can be installed or removed in seconds.

Price: 1-19 £16.90, 20-49 £15.40, 50+ £13.70

Evaluation kit £20 (SOFTLoK II, manual & software)
All prices excl. VAT and delivery

- For IBM PC, PS/2 and compatibles
- Uses parallel printer port
- Totally transparent to printer
- Secure data & password can be changed from your application software
- Cascadeable
- 240 bytes of secure read/write memory
- 8 byte (64 bit) password
- No programming adaptors required
- Easy to use SOFTLoK II setup software
- Routines ready to link with various compilers
- Easy to follow manual

I-MEX House, 40 Princess Street,
Manchester, M1 6DE, England,
Tel: 061 228 7379 **Fax:** 061 236 6890

CIRCLE NO. 961

Now we've got the language...

Some Problems for the C++ Community to Overcome If We Wish To See C++ Become a Widely Used Language.

So your software development team has learnt C++, understands classes and virtual functions, and has been converted to the benefits of object-oriented development. The time comes to start a large development project. So where do you go from here?

Accepted *industry* practice is to use project life-cycle methodologies to structure and control projects. Such projects pass through a number of stages on the route to completion: *high-level requirements definition and business cost justification, functional design, technical design, build and unit test* and *system test.*

Of course, the practitioners of such methodologies realise that blind adherence to a pre-defined sequence of stages does not guarantee a successful project - the talent of the development team is of course a vital factor. The use of methodologies does, however, provide an overall structure within which individual team members' creativity can be used to the full. It also provides the framework by which a project can be monitored against cost estimates - a vital factor for any commercial organisation. Finally, such methodologies provide useful analysis techniques such as *entity relationship diagrams, data flow diagrams or functional hierarchies.*

Herein lies the problem:

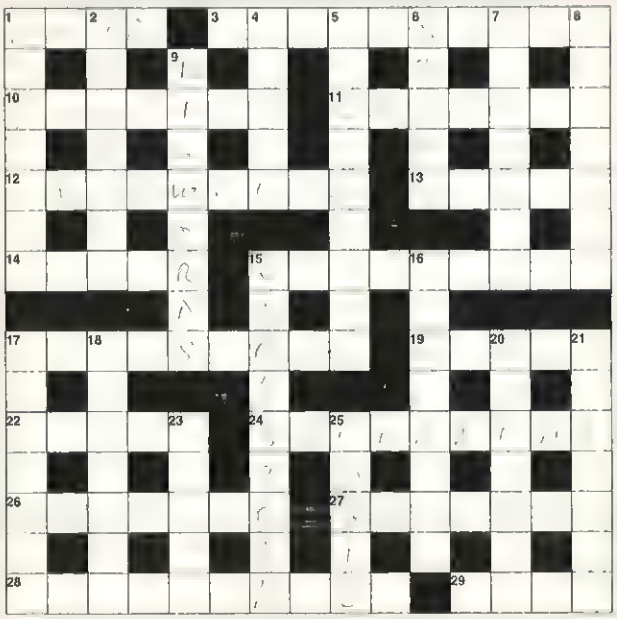
- Is the traditional waterfall cascade approach (outlined above) still valid in an object oriented environment? There is much current discussion about prototyping approaches, but how does one monitor (in a financial sense) such a prototyping approach?
- At which point in the life cycle does it become important to have an understanding of C++? Knowledge of C++ is obviously a prerequisite during technical design, but is it necessary during functional design?
- Exactly what is the relationship between established analysis techniques such as entity relationship diagrams/function hierarchies and class design? Intuitively there must be a relationship. Entity relationship diagrams contain information about the data (attributes) that is stored in business entities, and the (business) relationships between entities. Function hierarchies, which are generally cross referenced against entities, contain information about the functionality of the required system. Classes are an encapsulation of both data and functionality. There are clearly similarities between both approaches, especially when one considers the use of entity *hierarchies* (similar to class hierarchies?) in certain analysis techniques.
- What migration steps should be used to convert any existing analysis staff to an object-oriented approach? Indeed, is the analysis/programming split valid given this approach?
- What constitutes a *good* class design? What objective criteria can be applied to an object oriented design to ensure its integrity and completeness? Does the design have the capability to incorporate future enhancements cost-effectively - one of the main reasons for moving to an object-oriented approach in the first place?

These are some of the problems to which the C++ community must provide answers if we wish (and why shouldn't we) to see C++ become an established industry language. Answers on a postcard....

EXE

Subscription to the ECUG is £50 per annum. For more information about its activities, contact Rebecca Thomas on 071 253 5121, or write to ECUG, c/o City House, 190 City Road, London EC1V 2QH.

MAY .EXEWORD



ACROSS

1 Clearly a digital mouthful (4)
3 Use it to study the working range of a PC (10)
10 Not dreamily dealing with integers? (7)
11 Main raw material carrier (3,4)
12 Cover up when not in insert mode (9)
13 Lots of shots - five and none - follow little girl (5)
14 He makes life simple in recent festival without the start (5)
15 Principal support of dyno computer (9)
17 Loathing something of French software assessment (9)
19 Old sodium salt may get up my nose (5)
22 Be pleasant about Eastern relation in tree (5)
24 Using a bath to get things together on board (9)
26 Master from 29 confused by nothing (7)
27 Smart bug in the hardware (7)
28 Chatty group, maybe using phone, video or micros (10)
29 The other culture grown in smart systems (4)

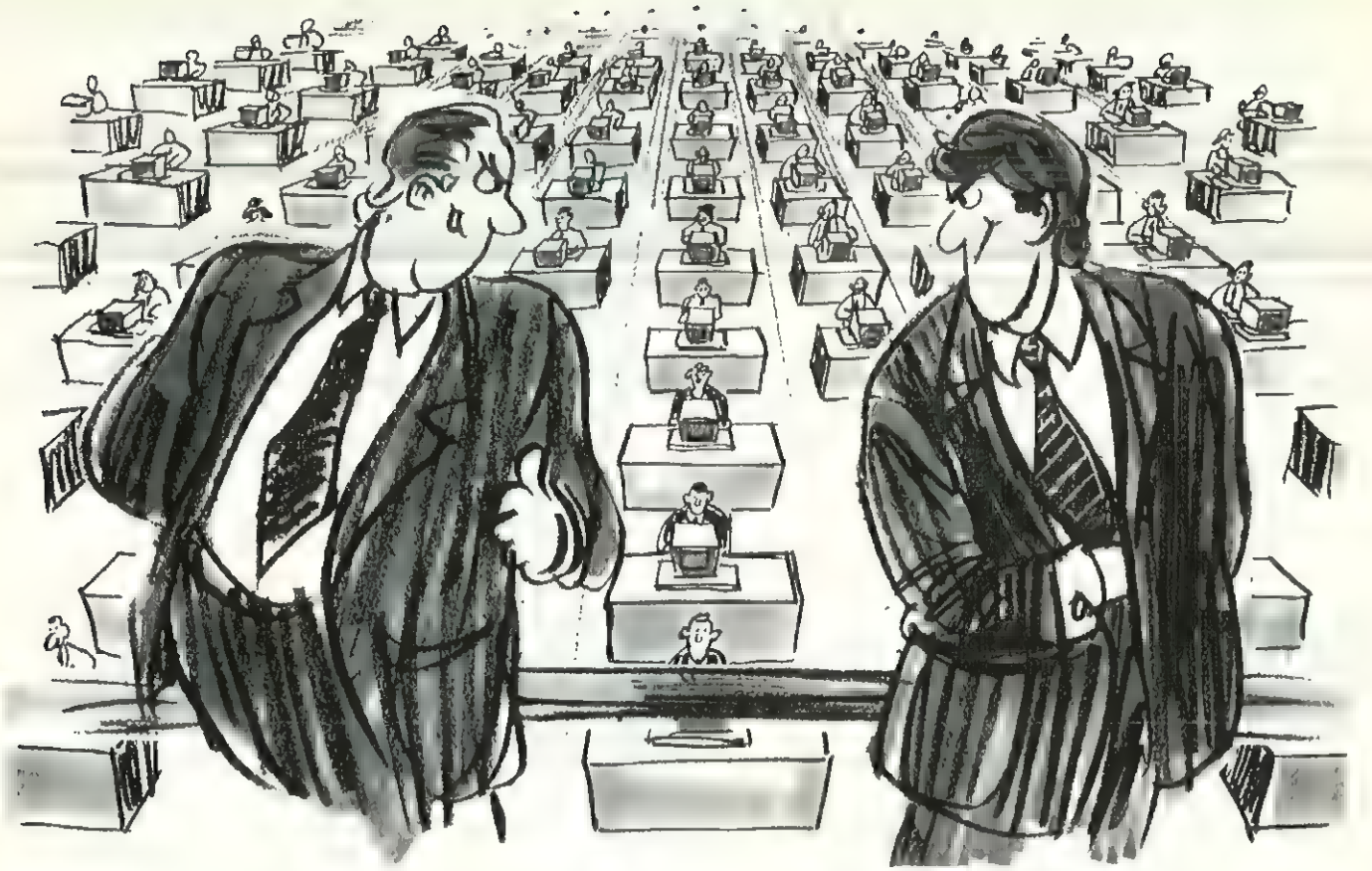
DOWN

1 Input pattern maybe observed by good 20 (7)
2 Folk ready to do a deal (7)
4 In full I'm starting two way feet (5)
5 Pub action after a break, eg with old file (9)
6 Parts of ladder in an algorithm (5)
7 Will made her mad in play (7)
8 Blow up like our field in the 80s (7)
9 64-bit units from a boaster (3,5)
15 Where to find current data in 15 ac, say (4,5)
16 Digitally told on the keyboard for instance (8)
17 Such good RAM keeps moving (7)
18 Neither one way or the other in that place (7)
20 Boozer avoided by Muslims? (7)
21 Fictional feet marking the keys maybe (7)
23 French come in for beginners' course without end (5)
25 Bit of wood on a chip may be the basis for non-fuzzy computing (5)

'.EXEWORD' compiled by Eric Deeson

C	O	N	N	E	X	I	O	N	■	S	C	R	A	P
R	■	E	■	X	■	B	■	O	■	O	■	O	■	A
A	N	T	I	C	■	M	O	D	U	L	A	T	O	R
S	■	W	■	E	■	S	■	U	■	U	■	O	■	T
H	O	O	T	E	R	S	■	L	I	B	E	R	T	Y
■	■	R	■	D	■	T	■	E	■	L	■	■	■	L
P	A	K	I	S	T	A	N	■	G	E	M	I	N	I
R	■	E	■	■	■	R	■	G	■	■	■	M	■	N
O	R	D	E	R	S	■	S	O	F	T	W	A	R	E
C	■	■	■	A	■	T	■	V	■	O	■	G	■	■
E	N	C	O	D	E	R	■	E	G	O	T	I	S	M
S	■	A	■	I	■	A	■	R	■	L	■	N	■	U
S	U	M	M	A	T	I	O	N	■	K	N	I	T	S
O	■	P	■	T	■	N	■	O	■	I	■	N	■	I
R	U	S	S	E	■	S	T	R	A	T	E	G	I	C

APRIL .EXEWORD

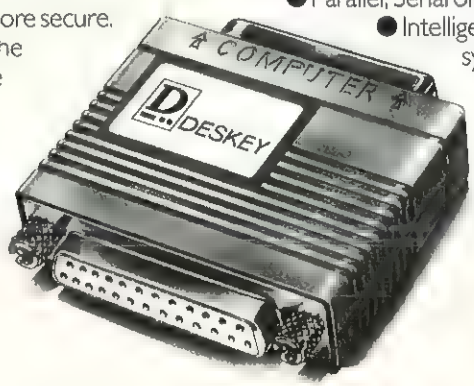


"You only need to buy one copy! They don't protect with DESkey."

DESkey – Software Protection from the inventors of the intelligent Dongle.

We have more experience and expertise in the design of software protection modules (dongles) than any other company in the UK. We sell products designed in house by our own engineers. More than 13 years experience in the design of dongles have gone into our current product range.

- DESlock© automatic .EXE or .COM file encryption system. No need for source or .OBJ files. Takes only a few seconds to protect your software – *now available for OS/2 and Windows.*
- 'Seedable' Pseudo Random Number Generator. Software and data encryption could not be more secure.
- Through Encryption. Data can be fed into the DESkey for on-line encryption. This keeps the encryption key hidden.
- Memory. Up to 240 bytes of memory split into 'Public' and 'Private' sectors. The 'Public' sector may be read from and written to at will. The 'Private' sector may be read at will, but writing requires your customer specific password.
- Down Counter. You program into the DESkey the number of times the program will run before stopping. This gives you the ability to sell 'goes' of your software rather than an open licence to use it forever – or even to send a fully working demonstration copy that will stop working after say, 10 goes.
- Variable Response Algorithm. This feature is similar to the well known 'public-key' algorithm and works in conjunction with an algorithm on the host computer. Makes any attempt at software emulation impossible.
- Secure Memory Read. Even if the same memory data is read repeatedly, the data returned from the DESkey never appears the same – this also makes the DESkey impossible to emulate.
- Parallel, Serial or Bus versions available.
- Intelligent Serial devices suitable for any operating system such as Unix, Xenix, OS/2, DOS etc. Works on any hardware including PC Networks, Mini Systems or even Mainframes.
- Free evaluation Kits.
- Guaranteed Exclusivity to all our customers.
- Fast Order Turn-round.
- Sold only through Data Encryption Systems Limited.



Edbrook House,
Cannington, Bridgwater,
Somerset, TA5 2QE

Telephone 0278 653456
Fax 0278 653300

Data Encryption Systems Limited

FAST
Federation Against Software Theft

Data Encryption Systems is a member of FAST

CIRCLE NO. 962

SIFIED...CLASSIFIED...CLASSIFIED...CLASS

To advertise in this section, please call Marc Warren on 081-994 6477. Fax 081-994 1533

Give *Real-Time* Projects a Head Start with the AMX™ low-cost high performance Multitasking Kernel

Features

Fast, reliable operation
Compact and ROMable
PC Peripheral support
DOS file access
C and Assembler supported
Preemptive task scheduler
Time slicing available
Configuration Builder
Intertask messages
Message exchanges
Dynamic operations:
- create/delete tasks, timers
- adjust task priorities
- memory allocation

Buffer Manager — Event Manager
Semaphore Manager — List Manager
Breakpoint Manager
Sample and Demo Programs
InSight™ Debug Tool for use with most popular debuggers

Targets

AMX 86 for real mode 80x86
AMX 386 for protected 80386
AMX 68000 for 680x0
AMX 80 for Z80, 64180 and 8085

No royalties
Source code included

For more information and a *free* demo disk contact:

ScotWare Ltd
45 Frederick Street, EDINBURGH, EH2 1EP
Tel: 031-225 4858
Fax: 031-220 3550

Real-time and database software specialists

WCSC PROFESSIONAL DEVELOPMENT TOOLS

DO NOT RE-INVENT THE WHEEL.

COMM-DRV-Professional Serial Communication libraries — $89.95 US.

COMM-LOG-True background XMODEM/YMODEM file transfer tool — $89.95.

MTASK-True preemptive multitasking kernel for MS-DOS — $89.95.

SCRN-INPUT-Character screen management package — $89.95.

TEL 713-498-4832
FAX 713-568-3334
ADDRESS: Willies Computer Software Co.
2470 S. Dairy Ashford Ste 188
Houston, TX 77077

PostScript Utilities

PostVue — £99 + VAT
A visual de-bugger for PostScript code. Watch your code execute, examine stacks, dictionaries and state variables. Set break points, warning levels, delay times.

PostPP — £49.00 + VAT
Un-scrambles PostScript code and lays it out in human readable form.

PPS — £49.00 + VAT
For all PostScript printers. Save paper with this versatile driver for ascii files. Output in columns, landscape, choose fonts, headers, line numbering etc.

Available from
QBS Software
10 Barley Mow Passage
London W4 4PH
Tel: 081 994 4842 Fax: 081 994 3441

KIBWORTH COMPUTER TRAINING

Expert individual attention makes best use of your time.

Full 5 day courses!

Personalised 'C' Tutorial — **£945**
C++/Practical object orientation — **£945**

Note that our prices include accommodation

68 Springfield Crescent
Kibworth Beauchamp
Leicester LE8 0LH

Tel: 0533-792653

---

brainy SYSTEMS (UK) LTD

Motherboards

386SX 25	£110
386SX 25 32K Cache	£135
386DX 33 128K Cache	£235
486DX 33 256K Cache	£475
486-33 EISA	£930
MULTI I/O (IDE 2S 1P 1G	£14

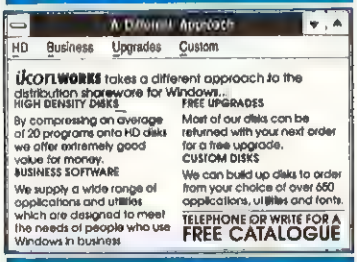
Systems

WE CAN BUILD TO ANY SPECIFICATION
PLEASE CALL FOR PRICES

Tel: 071 734 5783
Fax: 071 734 7256

---

windows SHAREWARE



ikonWORKS 22 Hope Street Lancaster LA1 3BQ 0524 39283

Ed — Soft as it Gets
the programmer's editor

QBS Software Limited,

10 Barley Mow Passage, London W4 4PH

Tel: 081-994 4842
Fax: 081-994 3441
BBS: 081-747 1979

© Soft As It Gets 1991

Programmers required

A successful software house requires top quality PC programmers to work on a graphics related Windows application.

Applicants must have a proven understanding of C and Windows but also be completely competent in 386/486 assembler as there are speed critical areas of code.

You would be joining a group of young, enthusiastic programmers producing a main stream application and become part of a small (a dozen programmers) but successful and growing software house.

Computer Concepts has been producing software for 10 years and is based in a large mansion in the Hertfordshire countryside. Facilities include tennis, snooker, swimming pool etc. Starting salary related to experience, but regularly reviewed so the right person has the opportunity for high rewards.

Assistance with accommodation possible.

CV to Charles Moir at;

Computer Concepts Ltd
Gaddesden Place, Hemel-Hempstead, Herts HP2 6EX
Tel 0442 63933 Fax 0442 231632

The Babelfish Paradox Data Driver for Clipper Summer '87 and 5.0x versions.

Call Now for a free Demo Disk.

QBS Software Ltd
10 Barley Mow Passage, London W4 4PH
Tel: 081 994 4842
Fax: 081 994 3441

BUGS WANTED!

A new BASIC User Group is looking for volunteer regional co-ordinators to help arrange regular meetings around the country.

If you have an interest or any experience with BASIC programming, please write to:

BUG
30 Osprey Close,
West Drayton, Middx UB7 7JE

PC Data Security Guide

Data Protection and Security for Personal Computers is a guide for support staff and managers. Everything you need to know about:

- Keeping viruses out of your company.
- Drafting and implementing a corporate data **security policy.**
- Persuading staff to take regular **backups** that are known to be reliable.
- Detecting and eradicating **software piracy.**

Includes comprehensive product guide and lots of trouble-shooting advice.

The price of £145 includes 90 days' **free telephone support** from the author.

Please call TTK Technical Publications on 081 995 9845 for details and a **free brochure.**

PostScript Library for C

Scripton

The successful Clipper library is now available for C programmers. There are over 50 functions in Scripton, allowing high definition text and graphics to be produced by PostScript printers. Embed letter-heads, logos and graphics in your Clipper systems.

- Improve report presentation
- Eliminate pre-printed notepaper

Available from — **Price: £155.00 + VAT**
QBS Software
10 Barley Mow Passage
London W4 4PH
Tel: 081 994 4842 Fax: 081 994 3441

The Secret of Successful Advertising is...

CLASSIFIED

To find out how you can reach 17,000 professional developers for as little as £85, call Marc Warren on 081 994 6477

SIFIED...CLASSIFIED...CLASSIFIED...CLASS

To advertise in this section, please call Marc Warren on 081-994 6477. Fax 081-994 1533

C++ Modeling & Simulation Class Library

Object Oriented Technology for Engineering, Manufacturing, Finance, Education

Meijin++™ 2.0

- Integration & Approximation
- Semi-Persistent Containers
- Discrete Event Simulation
- Numerical Analysis
- Queuing Systems
- Exceptions
- Statistics
- Plots

Royalty Free
115+ Classes
Sterling: $399.00
700 Pg User's Manual
MS-DOS/Ws, Unix, Sun
Documented Source Code
Professional Version Available

Network Integrated Services, Inc.
Tel: +1-714-755-0996
221 West Dyer Road
Santa Ana CA 92707
U.S. FAX
+1-714-433-2347

Fonts & PostScript

PostScript Interpreters: from £4 to £4,000
Fountain, GoScript, Freedom of Press, Ultrascript, PS Tutor, LincPage, NTG2000
PostScript LaserJet Upgrades from £239: Adobe, HP, Pacific Page, UDP TurboScript - HP/Canon
PostScript Utilities: ATM, PSFx, PSPlot, Books
Type 1 Fonts from £3 to £3,000: Public Domain, Adobe, ZSoft, Digi-Duit!, Watland Orb
HP Bitmap Fonts: ZSoft, Typografica, Digi-Duit!, Atech, DTS, Watland Orb, Tailor Made
Laser Printer Memory Upgrades: HP, Canon, IBM
Custom Fonts & Logos: most formats available
Custom HP Font Cartridges: from £90 (10 off)

Orchid '486 & Video

Motherboard + Fahrenheit 1280 runs up to 18 times faster than standard VGA: £960
Complete systems from £2469 with NEC 4FG

David Pollard Associates
Folly Bridge Workshops, Thames Street, Oxford OX1 1SU 0865 240048 Fax 792277

R&R Relational Report Writer

Database reports made easy for dBase, Clipper and compatibles. Also for Paradox, 123s and Quattro Pro.

Turn raw data into useful information using Lotus like commands and menus.

Available from
QBS Software Ltd
10 Barley Mow Passage
London W4 4PH
Tel: 081 994 4842
Fax: 081 994 3441

Call for free demo disk

WORKSTATION MEMORY

SUN 4 COMPATABLE MEMORY			
4/20/SCL		4Mb	£200
SPARCSTATION ELC		4Mb	£200.00
		16Mb	£675
4/40 (IPC)	8ONS	1Mb	£44.00
	7ONS	1Mb	£48.00
	8ONS	4Mb	£165.00
	7ONS	4Mb	£184.00
SPARCSTATION IPX		4Mb	£200.00
		16Mb	£675.00
4/60 SPARCSTATION 1		1Mb	£44.00
		4Mb	£165.00
4/65 SPARCSTATION 1+		1Mb	£44 00
		4Mb	£165 00
4/75 SPARCSTATION 2		4Mb	£165.00
4/110,150		1Mb	£44.00
4/260,280		8Mb	£1870.00
		16Mb	£2300.00
		32Mb	£3995.00
4/330		8Mb	£740.00
		16Mb	£1205.00
		32Mb	£2250.00
		64Mb	£4140.00
4/370,390		4Mb	£165.00
4/470,490		32Mb	£POA
		128Mb	£POA
SUN 3 COMPITABLE MEMORY			
3/50, 3/52		4Mb	£475.00
		8Mb	£765.00
		4MB	£305.00
SMM TOOL KIT			£35.00
3/60		1Mb	£44.00
3/80		1Mb	£44.00
3/75,110,140,150,160,180			
		4Mb	£750.00
		8Mb	£975.00
		12Mb	£1300.00
		16Mb	£1620.00
		20Mb	£1950 00
		24Mb	£2275.00
3/260,280,470,480		8Mb	£1870.00
		16Mb	£2300.00
		32Mb	£3885.00
3E120		4Mb	£505.00
		8Mb	£785.00
		12Mb	1075.00
SUN 386 WORKSTATION MEMORY			
386i,150		1Mb	£44.00
386i,150x,250		1Mb	£44.00
APOLLO DN MODELS			
DN2500	8ONS	1Mb	£44.00
	7ONS	1Mb	£48.00
DN3000, DN3010		2Mb	£280.00
DN3010A,3040,3500,3550,4000,4500,5500		4Mb	£370.00
		8Mb	£630.00
HP 400 SERIES			
HP9000/400,400dl,400t,425t,400s,433s		4Mb	£345 00
		8MB	£740.00
		16Mb	£1335.00
		32Mb	£2540.00
HP APOLLO 9000/425E			
		4Mb	£355.00
		8Mb	£725.00
		16Mb	£1200.00
HP APOLLO 9000/700			
		8Mb	£990.00
		16Mb	£1485.00
		32Mb	£3940.00
IBM COMPATIBLE MEMORY RS6000			
RS/320/320H,520,530,550,730,930,950		16Mb	£1250.00
		32Mb	£2290.00
		64Mb	£4390.00
APPLE COMPATABLE MEMORY			
Mac II,IIx,IIcx		1Mb	£42.00
SE,SE/30,PLUS,CLASSIC, LC,IIci,IIsi		4Mb	£170.00
Mac IIfx		4Mb	£200.00
		16Mb	£825.00
SILICON GRAPHICS COMPATIBLE			
PERSONAL IRIS 4D/20,25		1Mb	£44.00
		4Mb	£165.00
PROFESSIONAL IRIS 4D/50,70,80,85		1Mb	£44.00
		4Mb	£165.00
POWER SERIES 4D/120,210,220,240,280, 310,320,340,380		2Mb	£135.00
NEXT COMPATIBLE MEMORY			
NEXTcube,NEXTstation		1Mb	£42.00
		4Mb	£170.00

SABREBANNER
Tel 0532 854521
VISA, Access & MasterCard Taken
Fax 0532 853026

DCL COMPUTERS
COMPUTER PARTS

MOTHERBOARDS

286 AT 20MHz	£115.00
386 SX 20MHz	£250.00
386 AT 25MHz	£453.00
486 AT 25MHz	£1225.00

Inc. IDE, 2H, 2F, 2 serial parallel interface.

HARD DISKS

40Mb HD	£149.00
100Mb HD	£339.00

Other drives available.

For Price List & Details

Tel: 0902-335632
53 Wells Road, Penn,
Wolverhampton WV4 4BQ

HIRE INFORMATION TECHNOLOGY PLC

HIRE

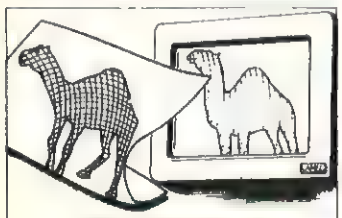
GUARANTEED VIRUS FREE

Hire I.T. is the ONLY PC Rental company that can Guarantee all systems are delivered Virus Free and stay that way throughout the hire, because Hire I.T. is the only PC rental company to load Anti-Virus software on every system we ship.

FOR COMPLETE PROTECTION FROM VIRUSES ONLY RENT FROM HIRE I.T.

071 490 1515

OLIVER HOUSE
51/53 CITY ROAD
LONDON EC1Y 2AD



PLOT THE DIFFERENCE

PLOTVIEW

is the simple low cost alternative to paper plots. Able to preview plot images on a graphics screen before printing

PLOTVIEW allows paperless plotting and reprinting avoiding delays improving efficiency

PLOTVIEW

Has up to 25 separate plot files previewable simultaneously on screen and a zoom function which allows any part of the plot to be inspected in detail

Easy to integrate PLOTVIEW is ideal for system builders and can be customised to suit their individual needs

Phone now for more details about PLOTVIEW the versatile tool that can really make a difference to your productivity.

CAMEL SERVICES LTD.
Telephone OXFORD (0865) 512678

JUST Computer BOOKS

- **Struggling with Sage?**
- **Dicing with DOS 5.0?**
- **Worried by WordPerfect?**

From beginner to expert
We have over 3000 titles to choose from.

Please phone or write for your FREE catalogue

111 Court Rd. Malvern
Worcs. WR14 3EF

☎ 0684-568095

Great News!

Object Oriented Real Time Kernel
for
Turbo Pascal
Now Available
from
JPS Graphics

For more information call
0244 821009
(Telephone or Fax)

JPS Graphics, 3 Glynne Street, Queensferry, DEESIDE, CH5 1TA

JPS Graphics

Computer MUSIC Systems Ltd

5-7 Buck Street,
London NW1 8NJ.
071-482 5224

Voyetra Sound Factory™

Sound Factory™ is a complete audio development kit for DOS or Windows 3, from Voyetra, the leaders in IBM music software. Containing drivers for a wide range of sound cards, digital audio boards & MIDI interfaces, Sound Factory™ gives developers as wide a market as possible. Call or fax us for further details.

We also stock a wide range of PC software and peripherals.

Specialists in Music Software

Software Engineer / Software Engineering Consultant

up to £17k plus car

C++ • Ada • Object Oriented Methods

Midlands Location

The successful applicant will join our small, friendly team with a varied mix of research, software development, training and consultancy projects for clients in automotive, defence and aerospace industries.

R&D projects will include motor racing related applications.

Applicants should be able to demonstrate experience with either Ada or C++ programming languages and related software engineering disciplines. The post would be ideal for a software engineer seeking to make a move into consultancy and training with the support of an established and experienced training and consultancy team. Although based at our new Midlands office, the post will provide ample opportunity to travel. Applicants should contact John Robinson at:

KBSL

KBSL, 1 Campus Road, Listerhills Science Park, Bradford, West Yorkshire BD7 1HR
Tel: (0274) 736895 Fax: (0274) 736553

WEST YORKSHIRE

'C' and SQL Assurance Programmers to work on IBM RS6000 under AIX *to £15,000*

Quality Assurance Analyst to test newly developed software and assist with program design/redesign *to £15,000*

Trainers for vertical market applications software, extensive travel involved *to £15,000*

BASIC Programmer to be trained subsequently in 4GL *to £15,000*

PC Technical Telephone Support to BIOS level *to £17,000*

Junior 'C' and 'C++' Programmer

For your next career move around West Yorkshire telephone Vincent Atherton on Leeds (0532) 504560 or write to:

AIREDALE RECRUITMENT

Realtex House, Micklefield Lane, Rawdon, Leeds LS19 6AX

AIREDALE RECRUITMENT

Technical Consultants

Network Controls International Ltd

We are the UK subsidiary of NCI Inc. We provide products and consultancy in the areas of retail banking and dealing room systems. We are currently looking to expand our team of technical consultants. The position will involve pre-sales analysis, system design and implementation, and support. The ideal candidates will have experience of most if not all of the following:

Dealing Room Systems **Retail Banking**
PCs (DOS, OS/2, Windows) **C** **PC Networks**

In addition to proven technical skills, we require individuals who are self-sufficient, have a high level of interpersonal skills and can respond to challenge. We expect an exceptional level of commitment from our staff and the salary will reflect this.

If you would like to be considered for the above position, then send your CV to: Liz Richardson, Network Controls International Ltd, Rowlandson House, 289-293 Ballards Lane, Finchley, London, N12 8NP

ASH ASSOCIATES

We specialise in the Recruitment of Software and Hardware Design Engineers for Real Time Applications including Graphics, Communications, Control/Robotics, Signal/Image Processing & Modelling, Real Time Financial Systems/Databases.

PICK OF THE BUNCH

CONTROL SYSTEMS DESIGN S/W & H/W DESIGN ENGINEERS SOUTH EAST to £25K

A rare opportunity to advance your career in Software or Hardware Design with this High Technology control systems designer.

This Small company has enjoyed considerable success and is now looking to the future by embarking on a complete re-design of these High Technology Systems.

They need both Software and Hardware Design engineers at Senior Level to take Project Responsibility for Graphics and Multi-Process Control Systems Design.

The Software Engineers experience will include 2&3D Graphics, GUI, Inter-Processor Comms, Case Tools and both High & Low Level languages gained ideally in a Real-Time Control Systems environment.

The Hardware Engineers will have similar experience of Control and Graphics Systems Design including ASIC and Signal Processing Devices, combined with a Real Commercial Flair and an Innovative Approach.

For this and Similar Opportunities Call James Hunt or Ron Cook NOW!

TEL: (0425) 475480 FAX: (0425) 480807

ASH associates

RECRUITMENT CONSULTANTS

3 Piper's Ash, Ringwood, Hants, BH24 1UF

PC APPLICS S/WARE DEVELOPMENT (Ref S1)
OXON: Proj Manager + Snr. S/Ware Eng with exp 'C', MS Windows, DOS, D/Bases eg. Paradox £20-40k - Also Team Ldr + Snr. S/Ware Test Eng with above PLUS S/Ware localisation tools & kits To £25k

TEAM LDR - PC's, NOVELL, UNIX (Ref D3)
BERKS: Small successful Co. needs above plus LAN Mgr, N/Work integration, PC app, S/W, 1st rate people skills. £25k package + car

PC HARDWARE & NOVELL - CUSTOMER SUPPORT (Ref D4)
BERKS: As above; strong skills in PC build and support also printers etc. and customer awareness. £17k package + car

OS/2, PM, C, CASE (Ref R1)
SURREY: Strong 'C' Developer with GUI's, SAA plats + strong product knowl. espec. CASE products. £17-26k

MS Windows, C, DOS, Comms (C++) (Ref R2)
E.BERKS: Geophysic. explor. Co. seeks S/Ware Eng with 2 yrs above; Image Proc, Novell/E'net.c£18k

APPLE MAC, C, SYSTEM S/WARE (Ref R3)
BUCKS OR DORSET: S/Ware Engs. & Proj. leaders in MAC S/ware, UNIX, WINDOWS, Servers, ISDN, Modems, ftp etc. £££s

INGRES ACCELL UNIFY SYBASE ORACLE (Ref R4)
MIDD'X/AVON/WILTS: Prime moves in Consultancy; 2 1 deg* + implem. Bids, Consult b/ground. >£30k

X WINDOWS, MOTIF, UNIX (Ref R5)
CAMBS/BEDS: Dynamic software Designers with in-depth X-Windows, C, GUI's.. real drive. £17-26k

RECRUITMENT CONSULTANCY (Ref R6)
W.BERKS/WILTS: Excellence buys Excellence! We're in the market for the best in the industry! £££s

MS WINDOWS, C, GRAPHICS (Ref R7)
WILTS or CITY: Strong product driven Designers with 2 yrs + WINDOWS, GUI's,DDE, UNIX. £18-27k

INGRES - V6.3 (ref D1)
CAMBS: Specialists with 2 yrs exp. of 'C' & embedded systems SQL, Commercial b/ground. £££s

PS/PC's - NETWORKS - APPLICATIONS (Ref D2)
MIDDX: Excellent DOS, WINDOWS, OS/2 Lan server skills also Lotus & M.S. products, 3270 comms. To £22k

0488 684944 (24 hours)

We are always interested in hearing from graduate calibre *Applications & Systems Designers to Consultant* level with skills in UNIX, RDB's, C, C++, GUI's SQL, CASE tools, XS-Windows, Motif, SSADM, OS/2, PC Comms, MS Windows, DOS etc.

RWA

FIRST CHOICE TO THE UK'S FOREMOST ENGINEERING ORGANISATIONS (SMALL AND LARGE) AND CONSEQUENTLY TO ITS 'TOP PROFESSIONALS'. TAP INTO OUR KNOWLEDGE NOW!

130 High Street, Hungerford, Berkshire RG17 0DL
Fax: 0488 685233

Software Development Translation and Test

Oxford & Cambridge £16,000-24,000

TEAM LEADER-TEST & QA

This key new appointment requires an individual with strong technical skills and ability to lead people and projects, often to tight deadlines. Candidates should be familiar with software Test/QA processes and have experience in PC databases, MS-Windows and GUIs. Superbase 4 experience would be particularly beneficial.

ANALYST/PROGRAMMER-SUPERBASE 4

Required for software house with increasing order book for high quality bespoke systems written in Superbase 4. Experience of the full development lifecycle is preferred although DML/RDBMS programmers with 1 years experience will be considered.

S/W TRANSLATION SPECIALIST

For this important role you will use development tools and kits to translate baseline PC software products into foreign languages. Knowledge of Windows, SDK and GUIs is highly desirable.

If you feel you are capable of rising to the challenge of these existing career opportunities please telephone **David Alcock** on **(0865) 742456/7** or send your CV to **Haybrook Appointments, Suite B, Regal Court, 112 London Road, Headington, Oxford OX3 9AU**

Haybrook APPOINTMENTS

PC Recruitment Specialists

RIVERSIDE HEALTH

DEPARTMENT OF MEDICAL ONCOLOGY
CHARING CROSS HOSPITAL
CANCER SERVICES

COMPUTER PROJECT/SYSTEMS MANAGER

Salary:
Grade 7 (£18099-£21621 incl of London Allowance)
plus ADP allowance on proven ability.

A Computer Manager is required for an expanding Cancer Centre. The job will involve working with the present systems and assisting the supplier in the production, implementation and then management of the new system due to be introduced in early 1993. Experience with Novell netware and expertise in programming (including dBase) is essential.

For informal discussion please contact
Dr G.Rustin, 081 846 1404 or Patrick McClean, 0263 861355.

For an application form and job description, please send an A4 self addressed envelope to:

The Recruitment Bureau,
Room 2503, Human Resources,
Charing Cross Hospital,
Fulham Palace Road,
London W6 8RF

or tel: 081 846 1480 (24 hour ansaphone).

Closing date: May 22nd 1992

Riverside is working towards equal opportunities

PEOPLE IN UNIX*

GET INTO GEAR

PREPARE FOR THE UPTURN

*Unix is a trademark of AT&T

ACTIS is seeking people with good technical experience of **OPEN SYSTEMS** which is still a growing sector of the IT market.

We have a number of instructions in the North and Midlands and we are looking for people with specific skills in:

Networking and Communications,
Relational Database/SQL
4 GLs, CASE, IPSE
C and WINDOWS
Accounts/Commercial and Scientific Programming
Training and Documentation
Technical Sales

We are seeking consultant, project management, development and support staff from graduate trainees to those with ten years experience.

To discuss these and other opportunities talk to **Honor Lindsey** on **(0204) 20200** or send your CV to **ACTIS RECRUITMENT 17 CHORLEY NEW RD, BOLTON BL1 4QR**

ACTIS

PEOPLE IN UNIX*

Are YOU looking for PC Systems Analysts, C Programmers, PC Support Analysts, Unix Gurus, S/W Development Engineers, Network Specialists, Object Oriented Systems Designers, Real Time Engineers, Database Developers, Application Support Engineers,Windows Programmers, CAD CAM Experts, Graphics Specialists.....and more?

LOOK NO FURTHER THAN THESE PAGES.

Call us on **081-994-6477** to find out more about this unique advertising medium.

Of course they all come with their experience in ADA, PASCAL, ORACLE, C, C++, WINDOWS, CLIPPER, MOTIF, SQL, CAD-CAM, ASSEMBLER, LANs, WANs, dBase UNIX.....

.EXE – We only provide the best!

---

the soft (corporation)

Specialists in relational databases, open systems and structured techniques.

ORACLE/SQL/ EMBEDDED SQL

Candidates with skills in SQL and embedded SQL are immediately required by clients involved in applications as varied as financial information systems to pan European manufacturing/ distribution and leasing systems. Strong SQL experience coupled with low-level language interfaces to 3GL's and screen painting methods are important. Training in the latest Oracle tools and methods given where necessary. The new and existing implementations are running under VMS, ULTRIX, UNIX and sunOS with some links to multi-user PC systems. Other database knowledge may also be of interest.

£12,000-35,000+
Ref SC/1/.exe

INGRES

Excellent opportunities exist for any INGRES development staff from Programmers up to Project Manager level. Candidates will be using the most advanced development tools available with companies investing in leading edge technology. INGRES/STAR, GUI's, SQL and OPEN SQL, decision support/application development tools, user interfaces and structured methods (SSADM, YOURDON, JACKSON), C, COBOL, are the priorities. Potential to work across varied hardware platforms is also a preference. Opportunities exist in Banks, Building Societies, Software and Systems Houses and Governmental Bodies.

£12,000-35,000+
Ref SC/2/.exe

SOFTWARE DEVELOPERS DATABASE DESIGN

SYBASE, INGRES, INFORMIX,ORACLE, UNIFY,CLIPPER, PRO IV, POWERHOUSE, RDB, ALL, SEACHANGE, PROGRESS, SB+, INFORMATION-and interfaces to 3GL's - C, C++, PASCAL and FORTRAN and even COBOL and BASIC! Our client companies require candidates with the following skills: Technical Software Development, Software Engineering, Pre/Post Sales, Database Design and Systems Consultancy. Areas of particular interest include: Distributed Systems, Object Oriented Databases, LANS, WANS, Graphics Image Processing, Windows, Formal Design Methodologies, CASE Tools, Expert Systems, Knowledge Engineering, Office Automation, Compilers, Low Level Design, Data Integration and varied application design. COMPUTER POLYGLOTS need apply!

£12,000-40,000
Ref SC/6/.exe

C/UNIX/MS-DOS

Software Houses and End Users in Manufacturing, Commercial, Scientific and Government application environments alike require excellent C skills. Low-level machine knowledge, operating systems internals, and use of debuggers/compilers are required. Software development experience is the key, and being able to deliver high performance, high quality, well specified software in competitive time scales. Opportunities vary from small software companies involved in expert systems, GUI's, Images Processing, GIS,EIS, Communications, Networking and Object Oriented Databases to large scale communication companies systems. Graduates through to senior software engineers/team leaders are required. UNIX internals and Kernel level knowledge are also very much in demand.

£14,000-35,000
Ref SC/3/.exe

COMMUNICATIONS/ NETWORKING

Developments as varied as Voice-activated relational database recognition systems, Protocol enhancement at transport, session and presentation level and World-wide communications systems utilising LAN's and WAN's across different hardware platforms are currently available. Experience of ETHERNET, TCP/IP, NFS, X25, X400, X500 in a UNIX, VMS, sunOS and also fault tolerant environments are required. Some exposure to structured methods and other leading edge technology would be a bonus, though training will be given. Knowledge of industry standards and committees is also relevant at more senior levels. UNIX Kernel knowledge is at a premium.

£12,000-£35,000
Ref SC/4/.exe

X WINDOWS/ MS-WINDOWS/MOTIF

Graduates (1 year+) to Senior Software Engineers with an interest in advanced development environments need apply for varied positions with companies dedicated to leading edge technology. A mixture of the following skills in a development environment are preferred: GUI's, Document Image Processing, OCR technology, Client/Server applications and WYSIWYG techniques coupled with experience of UNIX, sunOS, MS-DOS, or ULTRIX/VMS and Interfaces to relational (primarily INGRES or SYBASE) and Networking (TCP/IP, NFS, X400) technology. Application areas vary from Retail, distribution to finance and Software Vendors to End User environments.

£12,000-35,000 + BENEFITS
Ref SC/5/.exe

SOFTWARE ENGINEERS

Clients-End-Users, Software Vendors and Software Houses dedicated to strategic implementation of leading edge technology and integration of applications across different hardware and operating systems platforms require candidates to degree level with a scientific/technical development bias and a 1-3 years' experience. There are two main options:-

TECHNICAL DEVELOPMENT: Continued use of UNIX, VMS, MS-DOS, C, Windows, Pascal, C++, Ada, Prolog, OOPS, Networking and Communications with companies offering technology based careers and management responsibility.

COMMERCIAL DEVELOPMENT: Using technical skills already developed, but offering opportunities to apply analysis and design skills rather than remain a 'technical guru'. Please call to discuss your particular career path, growth and potential.

£12,000-25,000 + BENEFITS
Ref SC/7/.exe

SYBASE/UNIFACE

Due to the continued expansion of the market for RDBMS as a multi purpose software development tool and particularly the adaptability of SYBASE, its popularity is growing in manufacturing, scientific, industrial, commercial and financial application areas. Clients throughout the UK and Europe require SYBASE experience in a VMS, UNIX and even PC environment. Again interfaces to 3GLs, Structured Methods and Communications Protocols are also important. Please call to discuss the varied opportunities at all levels, including setting up new development departments.

£12,000-40,000 + BENEFITS
Ref SC/8/.exe

INFORMIX/CLIPPER

Clients involved in small to medium database projects (10 to 35 users) utilising products like INFORMIX, CLIPPER, DBASE IV and FOXBASE require Programmers Analyst/ Programmers with 4GLs, C, Pascal, Turbo Pascal, Quickbasic etc... or some of these skills for new and existing developments. Potential to get involved in Networking projects and in some cases larger scale projects involving cross-training also exist.

£12,000-22,000
Ref SC/9/.exe

STOP PRESS

Pre/Post Sales UNIX Systems London.
SQL Systems Consultant - Berk/Surrey/London.
UNIX Systems Administrators - All Areas.

CONTRACT AND PERMANENT POSITIONS ARE UK WIDE, SO PLEASE TO DISCUSS YOUR PARTICULAR SKILLS AND REQUIREMENTS.

The Soft Corporation Chancery House 319 City Road London EC1V 1LJ Tel: 071 609 5501 Fax: 071 700 5787

ADVERTISERS INDEX

STOB - Testing away

Verity Stob has been benchtesting 25 PC Magazines in the .EXE labs.

British PC Person. 324 pages. Main features: comparison of 23 VGA colour monitors, review of the new WordStarPerfect for Windows word processor ('we liked the way that WordStarPerfect takes advantage of the Windows environment, without betraying its DOS origins... With its speedy response and its QuickTypoBar feature, how can this package fail?'), an overlong April Fool joke item purporting to compare the comparative features of 35 mouse mats, and a rich selection of regular columnists and smaller reviews. We particularly admired Danny Rat's technical column: Why I don't have to write anything this month because fortunately you've sent in enough stuff.

Noise made when dropped on carpeted floor from a height of six feet: *ffflonk!*

American Corporate Computer Type, UK Edition. 478 pages. A newcomer to the news-stands, this issue of ACCT UK came with a free 3.5" disk of 'white-hot DOS utilities', including a menu-driven replacement for the DIR command (unfortunately in French, 'Les fiches sont...'), a dBASE II compatible card index system and a crash-prone demo version of a game called 'Gerbils'. The magazine was divided between 390 pages of advertisements, many of them different, and 88 pages of editorial. The main feature was a comparative review of 55 different brands of PC keyboard ('if it is to make any impact in the PC keyboard market, Keitso really must do something about the positioning of the Caps Lock LED on its 112-key range'). Support features included a review of the new WordStarPerfect for Windows word processor ('now that Windows is established as the corporate operating system of choice, WordStarPerfect really needs to make up its mind whether it is going to support the new environment or drift into retirement under DOS... this bodge of a word processor is neither one thing or the other, notwithstanding the inclusion of stupid gimmicks like the QuickTypoBar') and a column by Bill Hack, the Fearless Voice of Corporate Computing ('now that OS/2 2.0 has arrived, questions must hang over Windows's status as the corporate operating system of choice').

Noise made when dropped from a height of six feet: *ker-splatt!*

PC-Fuss. 320 pages. PC-Fuss also comes with a disk of 'Brill Utils and K-k-krazy games' sticky-taped to its cover; programs included were 'Disk Surgeon', a FAT table repair facility that copes quite well with pre DOS 4 floppy disks, 'MyMate', a memory-resident ASCII editor which requires 100 KB of conventional RAM to run, and 'Piggy the Porcupine', the demo version of an action arcade game from the people who brought you Wally the Wombat, Ernest the Earwig, Toppy the Tapeworm etc. PC-Fuss's main feature is a comparative review of 104 text-mode based word processors ('because whatever you say about Windows, it's clear that good ol' DOS is going to be with us for some time to come'). Also reviewed is the WordStarPerfect for Windows word processor ('now that everybody uses Windows all the time, it's good to see that WordStarPerfect has finally made the leap into the 19th Century and left slimy old DOS behind it').

Noise made when dropped: *whooomphh.*

The Editor has spiked Ms Stob's copy covering the remaining 22 magazines, on the grounds that only bananas would wish to read a review of so many drearily similar products. We believe that the market for such bananas is already amply catered for.

EXE

1 Golden Square, London W1R 3AB
Tel: 071-734 8434 Fax: 071-734 8379

Senior Software Engineer
Midlands **£18K**

You will be expected to produce software packages usually in Assembly level language for incorporation into this company's product range. You will be working by yourself or on small team projects. The kind of skills that are required are: good Assembler (z80 and linker), MASM 8086 Assembler and linker, MS DOS and one high level language preferably C, Basic or Pascal. You must have at least three years experience and have a flexible approach to work.

OSF Motif X-Windows UNIX Developers
Cambs **£23K**

This renowned Company that is heavily involved with UNIX Windows and utility development is desperate for some extremely skilled individuals. You should have a good development background and be used to dealing with clients/end-users specific tailoring requirements. There should be some international travel involved, (User Group and Conferences etc) and you should feel self sufficient in your own ability. Being a graduate is not necessarily a prerequisite, just being an able developer is...

SQL RBT or Windows SQL Specialist
Berks **Car & up to £30K**

This Company is a renowned international software house that requires two such individuals. You must have a proven development record within at least one major project. In return, you will have a vigorous and varied role, throughout Europe and GB; consulting, tailoring and supporting the company's products and developments; would suit a coder who wants to have a more consultative role. You should be a graduate, or high flying individual with A Levels at the least.

UNIX Motif Realtime Dealing Systems C
London **Benefits and up to £25K**

This solutions house have a tremendous reputation in the stock market/'broking trade. the product is proven and a success - the company is looking to employ a new project team and requires a calibre developer, preferably with a minimum of two years experience and probably a Computer Science graduate.

Windows Datacomms Novell Connectivity
Middx **Car up to £27K**

This company has been a proven success story since setting up their European office here two years ago. They are involved in AS400 and mainframe client/server style connectivity, for PC Windows applications. You should have worked in Comms/Network environment and feel confident in your 3270 skills.

4GL PROIV Complex RDBMS Multiuser
London **Up to £20K**

Pick, mainframe, UNIX platforms or any other multi-user experience would be extremely important to the company. A broad sweep of skills for this information/image processing/satellite company is a prerequisite. Structured methodologies are a must and a willingness to learn new skills is essential.

Group MIS Manager UNIX Kilostream
Sheffield + Europe **Benefits, £Good**

Good European languages; French or Spanish, ten years experience and networks TCP/IP or 3COM products could be useful. It could be envisaged that some European travel will be a regular element to the job. Typical backgrounds would include development and some technical support, also some HP3000 platform work could be useful, but not essential.

Realtime Comms Development Assembler C
Kent **Up to £22K**

6800, 6809, OS9 and comms protocol experience; SDLC & PLC. Other work will include development on Xenix emulators, comms analysers and the like. This development would appeal to a Senior Software Engineer who wishes to combine classic comms work with C coding and Realtime applications. the company is involved with process control software and is extremely successful in the arena.

C Windows PC Smalltalk C++
W.London **£Good**

Development department of a large Japanese hardware company is getting heavily involved in a GUI bundling exercise. They want an individual who has a solid computer science/software engineering graduate background - second jobber, minimum experience is 2 years in development.

Mainframe Products in a PC Environment
Berks/Hants **£Good, Car, Benefits**

Product support role for PC company emulating MVS development; utilities, operating systems, tools, development languages and 4GLs. You should have had good experience in all of these and be able to "translate" their applications to a PC Lan development environment. A challenging role for the right individual, it is envisaged that the minimum age requirement would be at least 28, you should have good mainframe system level skills.

Sybase Oracle Ingres Informix PC or UNIX
Home Counties **Up to 35K + Benefits**

A major player in the banking and finance software sector is looking for high calibre developers. You must be used to a high degree of flexibility and have a structured development background - Yourdon, Jackson etc. The rewards will be evident and a good career path with European travel are a certainty.

IF ANY OF THESE POSITIONS ARE AT ALL SUITABLE, OR EVEN IF YOUR SKILLS ARE NOT QUITE WHAT IS REQUIRED, DO NOT HESITATE TO GIVE MIKE DEARING A CALL ON 071-734 8434 OR AFTER HOURS ON 081-767 1003. LOOK FOR REGULAR UPDATES ON ORACLE TELETEXT.

THE BEST OS/2 2.0 TRAINING AND CONSULTANCY

Number One in OS/2

QA Training is the leader in OS/2 2.0 Technical Training and Consultancy.

We were the first company in the world to give training in Presentation Manager; OS/2's advanced Graphical User Interface.

We were the company chosen to supply OS/2 2.0 training and education materials to IBM.

We have courses covering OS/2 2.0 Database, Communications and LAN products for both programmers and support personnel.

And our consultants' expertise can be called in at any stage of an OS/2 project, from planning to implementation.

So, if you are interested in getting the best out of OS/2, why not join those who have discovered that QA is the one to help you.

For more information, please complete and return the coupon today or call Samantha Trinder on 0285 655888.

MIGRATING TO OS/2 2.0

OS/2 TECHNICAL OVERVIEW 1 day

MIGRATING OS/2 APPLICATIONS TO 32-BITS 2 days

OS/2 2.0 PROGRAMMING COURSES

OS/2 PROGRAMMING 4 days

OS/2 PRESENTATION MANAGER PROGRAMMING 5 days

ADVANCED OS/2 PRESENTATION MANAGER PROGRAMMING 5 days

IBM OS/2 DATABASE MANAGER PROGRAMMING 4 days

OS/2 APPC PROGRAMMING 4 days

OS/2 DEVICE DRIVERS 5 days

OS/2 SUPPORT COURSES

INTRODUCTION TO OS/2 1 day

USING OS/2 QUERY MANAGER 2 days

SUPPORTING OS/2 4 days

SUPPORTING OS/2 COMMUNICATIONS MANAGER 3 days

SUPPORTING OS/2 DATABASE MANAGER 3 days

OS/2 LANS

OS/2 LAN PROGRAMMING 4 days

SUPPORTING IBM OS/2 LAN SERVER 4 days

CONSULTANCY

FEASIBILITY

DESIGN STANDARDS

DESIGN WRITING

PROJECT MANAGEMENT

PROBLEM SOLVING

IBM

Suppliers of OS/2 training and education materials to IBM

Mail to QA Training Ltd, Cecily Hill Castle, Cirencester, Gloucestershire GL7 2EE, UK.

Tick here for full details:

TRAINING: ☐ OS/2 ☐ Windows ☐ Networks ☐ Unix/Aix ☐ C and C++ ☐ Object-Oriented

☐ CONSULTANCY SERVICES

☐ DEVELOPMENT TOOLS

NAME _____ TITLE _____

COMPANY _____

ADDRESS _____

_____ TEL. NO. _____

CIRCLE NO. 966

QA TRAINING

Number One in OS/2

AEXE4